O'REILLY®

TURING 图灵程序设计丛书

面向数据科学家的实用统计学

Practical Statistics for Data Scientists

系统梳理数据科学中重要的统计学概念，演示统计学方法在数据科学中的应用



[美] 彼得·布鲁斯 安德鲁·布鲁斯 著

盖磊 译

 中国工信出版集团  人民邮电出版社 POSTS & TELECOM PRESS

译者介绍



盖磊

计算机博士。目前供职于研究机构，主要从事数据分析相关研究工作。具有扎实的统计学理论基础和丰富的实践经验，并已完成多部科技图书的翻译。

数字版权声明


图灵程序设计丛书

面向数据科学家的实用统计学

Practical Statistics for Data Scientists
50 Essential Concepts

[美] 彼得·布鲁斯　安德鲁·布鲁斯　著
盖磊　译

Beijing • Boston • Farnham • Sebastopol • Tokyo　O'REILLY®

O'Reilly Media, Inc.授权人民邮电出版社出版

人 民 邮 电 出 版 社
北　京

图书在版编目（CIP）数据

面向数据科学家的实用统计学 / (美) 彼得·布鲁斯 (Peter Bruce)，(美) 安德鲁·布鲁斯 (Andrew Bruce) 著；盖磊译. -- 北京：人民邮电出版社，2018.10
（图灵程序设计丛书）
ISBN 978-7-115-49366-8

Ⅰ. ①面… Ⅱ. ①彼… ②安… ③盖… Ⅲ. ①统计软件 Ⅳ. ①C819

中国版本图书馆CIP数据核字(2018)第212556号

内 容 提 要

本书解释了数据科学中至关重要的统计学概念，介绍如何将各种统计方法应用于数据科学。作者以易于理解、浏览和参考的方式，引出统计学中与数据科学相关的关键概念；解释各统计学概念在数据科学中的重要性及有用程度，并给出原因。

本书适合数据科学从业人员阅读。

◆ 著　　[美] 彼得·布鲁斯　安德鲁·布鲁斯
译　　盖　磊
责任编辑　岳新欣
责任印制　周昇亮

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京　　　　印刷

◆ 开本：800×1000　1/16
印张：14.75
字数：349千字　　2018年10月第1版
印数：1－3 000册　　2018年10月北京第1次印刷

著作权合同登记号　图字：01-2018-3440号

定价：89.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京东工商广登字 20170147 号

版权声明

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 *Make* 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过图书出版、在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——*Wired*

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

目录

本书献给我们的父母维克多·布鲁斯和南希·布鲁斯，以纪念他们。正是他们培养了我们对数学和科学的热爱。也献给我们早年的导师约翰·图基和朱利安·西蒙，以及我们毕生的朋友杰夫·沃森。他们启发我们以统计学作为一生的职业。

前言

本书所面向的读者是那些在一定程度上熟悉 R 编程语言，并具有一些统计学知识（即便是碎片化的知识，或是短期接触过统计学）的数据科学家。作为本书的作者，我们都是从统计学领域迈入数据科学领域的，因此对统计学在数据科学中可做的贡献有所了解。同时，我们也十分清楚传统的统计学教学的局限所在，即统计学作为一门学科已经有 150 多年的历史了，大多数统计学课本和课程都表现出远洋轮船般的动量和惯性，很难有所改变。

本书有两大目标：

- 以易于理解、浏览和参考的方式，引出统计学中与数据科学相关的关键概念；
- 解释各个统计学概念在数据科学中的重要性和有用程度，并给出原因。

本书的独到之处

> **主要术语**
>
> 数据科学融合了多门学科，包括统计学、计算机科学、信息技术和一些特定领域的研究。因此，同一个概念可能会使用多个不同的术语表述。本书将使用类似此处的格式，突出显示各个主要术语及其同义词。

排版约定

本书将使用如下排版约定。

- **黑体字**
  用于标识新的术语。
- 等宽字体（`constant width`）
  用于标识程序清单，以及段落内引用的程序元素，例如变量、函数名称、数据库、数据类型、环境变量、程序语句和程序语言关键字等。

- 等宽粗体（**`constant width bold`**）
  表示应由用户逐字输入的命令或其他一些文本内容。

- 等宽斜体（*`constant width italic`*）
  表示文本应被替换，替换内容由用户提供，或取决于上下文。



此图标表示一个知识点或一条建议。



此图标表示一处通用注解。



此图标表示一条警告或警示。

使用代码示例

本书的补充材料（即示例代码、练习等）可从 https://github.com/andrewgbruce/statistics-for-data-scientists 下载。

本书旨在帮助你更好地完成工作。一般来说，只要是本书提供的示例代码，你都可以用于自己的程序和文档。除非你需要大规模地使用本书的代码，否则无须联系作者以获得许可。例如，你在编写代码时使用了书中的几处代码是不需要获得许可的，但销售或分发 O'Reilly 图书中的 CD-ROM 则需要获得许可。在回答问题时引用本书内容和示例代码不需要获得许可，但在产品文档中整合本书中的大量示例代码需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明通常包括书名、作者、出版商和 ISBN。例如："*Practical Statistics for Data Scientists* by Peter Bruce and Andrew Bruce (O'Reilly). Copyright 2017 Peter Bruce and Andrew Bruce, 978-1-491-95296-2."

如果你认为自己对示例代码的使用超出了合理使用的范围或是上面介绍的许可范围，可随时通过电子邮件 permissions@oreilly.com 联系我们。

Safari® Books Online

Safari Books Online 是一个按需提供服务的数字图书馆，所提供的图书和视频来自于在技术和商业上处于世界领先地位的作者。

Safari Books Online 已被专业技术人员、软件开发人员、Web 设计师以及商业和专业创意人员使用，成为科学研究、解决问题、学习与认证培训的主要资源。

Safari Books Online 为企业、政府、教育机构和个人提供了一系列的计划和定价。

会员访问一个功能完备的数据库检索系统，就可以获得上百家出版商的上千种图书、培训视频和预发行手稿，其中包括 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 等。有关 Safari Books Online 的更多信息，可在线访问。

联系我们

请把对本书的评价和问题发给出版社。

美国：

> O'Reilly Media, Inc.
> 1005 Gravenstein Highway North
> Sebastopol, CA 95472

中国：

> 北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
> 奥莱利技术咨询（北京）有限公司

本书页面提供了本书的勘误表、例子及其他信息，网址为 http://shop.oreilly.com/product/0636920048992.do。[1]

对 O'Reilly 图书的评论和技术问题，可以发送电子邮件到 bookquestions@oreilly.com。

关于 O'Reilly 图书、课程、会议和新闻等更多内容，参见网站 http://www.oreilly.com。

在 Facebook 上关注我们：http://facebook.com/oreilly。

在 Twitter 上关注我们：http://twitter.com/oreillymedia。

在 YouTube 上关注我们：http://www.youtube.com/oreillymedia。

致谢

作为本书的作者，我们希望在此感谢对本书出版提供过帮助的许多人。

数据挖掘公司 Elder Research 的首席执行官 Gerhard Pilcher 审读了本书的初稿，并做出了详细而有用的修正和评论。同样，SAS 的统计学家 Anya McGuirk 和 Wei Xiao，以及同是 O'Reilly 作者的 Jay Hilfiger，也对初稿提出了有益的反馈。

注 1：本书中文版的勘误请到 http://www.ituring.com.cn/book/2066 查看和提交。——编者注

在 O'Reilly 出版社方面，Shannon Cutt 给予我们鼓励并适当地敦促我们，还在出版流程上提供了指导。Kristen Brown 使本书得以顺利地推进到生产制作阶段。Rachel Monaghan 和 Eliahu Sussman 耐心而又细致地修改了本书的书稿。Ellen Troutman-Zaig 为本书做了索引。我们还要感谢 O'Reilly 发起本书项目的 Marie Beaugureau，以及 statistics.com 讲师兼 O'Reilly 作者 Ben Bengfort，正是他将我们介绍给了 O'Reilly。

Galit Shmueli 曾与 Peter 合著过其他图书，并且多年来一直与 Peter 保持交流。这种交流使我们乃至本书都受益匪浅。

最后，我们要特别感谢 Elizabeth Bruce 和 Deborah Donnell，没有她们的耐心和支持，就不会有这本书。

电子书

扫描如下二维码，即可购买本书电子版。



第 1 章

探索性数据分析

在过去的一个世纪中，统计学作为一门学科得到了长足的发展。概率论是统计学的数学基础，它基于托马斯·贝叶斯、皮埃尔·西蒙·拉普拉斯和卡尔·高斯等人的工作，在 17 世纪至 19 世纪期间形成并发展。与概率论的纯理论本质不同，统计学是一门应用科学，关注的是数据的分析和建模。现代统计学是一门严谨的科学，其根源可上溯至 19 世纪末的弗朗西斯·高尔顿和卡尔·皮尔逊。20 世纪初，罗纳德·艾尔默·费希尔成为现代统计学的先驱之一，他提出了**实验设计法**和**最大似然估计**等重要概念。不少其他统计学概念在很大程度上也深深地植根于数据科学中。本书的主要目标就是帮助你理解这些概念，并阐明这些概念在数据科学和大数据的背景下是否依然重要。

本章的重点是探索数据，这是所有数据科学项目的第一步。**探索性数据分析**（EDA）是统计学中一个相对新的领域。经典统计学几乎只注重**推断**，即从小样本得出关于整体数据的结论，这往往是一个复杂的过程。1962 年，约翰·图基（图 1-1）发表了一篇著名的论文“The Future of Data Analysis”，由此引发了对统计学的重构。在论文中，图基提出了他称之为**数据分析**的一门新学科，并将统计推断包括于其中，由此建立了与工程和计算机科学界的联系［他提出了术语**比特**和**软件**，其中“比特”（bit）是“二进制数字”（binary digit）的缩写］。出乎意料的是，这一初始理念被延续了下来，并成为了数据科学的基础之一。图基编著并在 1977 年出版了 *Exploratory Data Analysis* 一书，该书开创了探索性数据分析这一研究领域，现已成为一本经典图书。

随着计算能力和数据分析软件可用性的提高，探索性数据分析的发展已远超其最初的范围。该学科的主要驱动力来自于新技术的快速发展、更多及更大规模的可访问数据，以及定量分析在多个学科中更广泛的应用。斯坦福大学统计学教授戴维·多诺霍曾撰写过一篇很好的文章，文中将数据科学的起源追溯为图基在数据分析领域所做的开创性工作。多诺霍教授在本科期间曾得到图基的指导，该文是他基于自己在美国新泽西州普林斯顿召开的

图基教授百年纪念研讨会上的演讲[1]而撰写的。

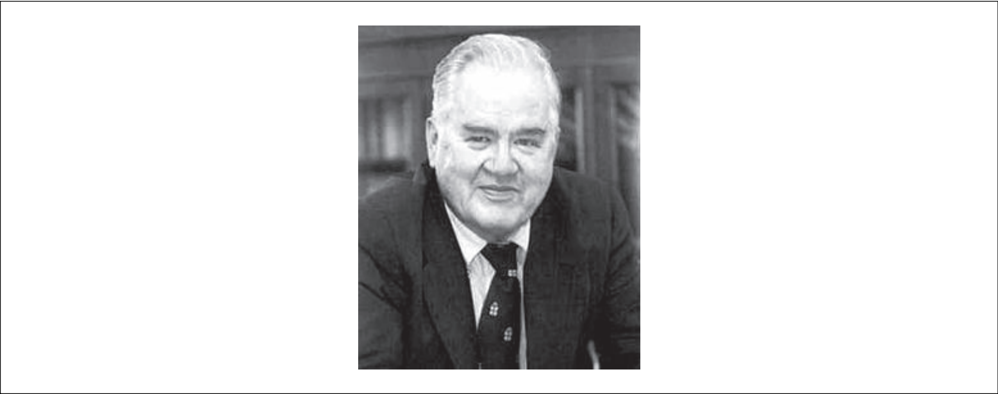


图 1-1：约翰·图基，著名统计学家，他在 50 多年前提出的理论构成了数据科学的基础

1.1 结构化数据的组成

数据的来源非常丰富，例如传感器的测量值、事件、文本、图像和视频等，并且**物联网**正在涌出大量信息流。这些数据大多是非结构化的。例如，图像由一系列像素点组成，每个像素包括了红、绿、蓝三原色信息；文本是单词和非单词字符的序列，常以章节、子章节等形式组织在一起；点击流是用户在 App 或 Web 页面上的动作序列。事实上，如何将大量的原始数据转化为可操作的信息，这才是数据科学所面对的主要挑战。要使用本书中介绍的统计学概念，就必须将非结构化的原始数据结构化（就像是从关系型数据库中取出的数据那样），或者出于研究目的采集数据。

主要术语

连续型数据

数据可在一个区间内取任何值。

同义词：区间数据、浮点型数据、数值数据

离散型数据

数据只能取整数，例如计数。

同义词：整数型数据、计数型数据

分类型数据

数据只能从特定集合中取值，表示一系列可能的分类。

同义词：枚举数据、列举数据、因子数据、标称数据、多分支数据

注 1：Donoho, David.“50 Years of Data Science”(2015).

二元数据

一种特殊的分类数据，数据值只能从两个值中取其一（例如 0 或 1，True 或 False）。

同义词：二分数据、逻辑型数据、指示器数据、布尔型数据

有序数据

具有明确排序的分类数据。

同义词：有序因子数据

结构化数据有两种基本类型，即数值型数据和分类数据。数值型数据有**连续型**和**离散型**两种形式。风速、持续时间等是连续型数据，而某一事件的发生次数则是离散型数据。**分类数据**只能取一系列固定的值，例如，电视屏幕的类型可以是等离子体、LCD 或 LED 等，美国各州的名称包括阿拉巴马州、阿拉斯加州等。**二元数据**是一种重要且特殊的分类数据，该类数据的取值只在两者中择其一，例如 0 或 1、是或否、True 或 False 等。**有序数据**是另一种有用的分类数据，该类数据是按分类排序的，例如数值排序（1、2、3、4 或 5）。

我们为什么要关心数据类型的分类呢？事实表明，在数据分析和预测建模中，数据类型对于确定可视化类型、数据分析或统计模型是非常重要的。R 和 Python 等数据科学软件也使用数据类型去改进计算性能。更重要的是，变量的数据类型决定了软件处理变量计算的方法。

对此，软件工程师和数据库编程人员可能会产生疑问：为什么我们在数据分析中也需要了解**分类数据**和**有序数据**呢？毕竟，分类数据只是一组文本值（或数值），数据的内部表示会被后台的数据库自动处理。但是，相比于文本表示，将数据显式地标识为分类数据的确具有如下优点。

- 如果我们明确输入的是分类数据，那么软件就可以据此确定统计过程的工作方式，例如图表生成或模型拟合。具体到 R 和 Python 中，有序数据可用 `ordered.factor` 表示，这样用户指定的顺序就能保持在图、表和模型中。
- 可以优化存储和索引，如同在关系型数据库中那样。
- 限定了给定分类变量在软件中的可能取值，例如枚举类型。

第三个优点可能会导致一些令人意想不到的行为。R 语言的数据导入函数（例如 `read.csv`）默认将一列文本自动转换为因子（`factor`）。随后操作该列数据时，会假定所允许的值局限于先前已导入的值。此时赋值一个新的文本值，将会触发警告，并生成 `NA`（即缺失值）。

本节要点

- 在软件中，数据通常按类型分类。
- 数据类型包括连续型数据、离散型数据、分类数据（其中包括二进制数据）和有序数据。
- 数据分类为软件指明了数据的处理方式。

拓展阅读

- 数据类型有时会令人困惑，因为各类型间会有一些重叠，而且不同软件的数据分类可能各有不同。R Tutorial 网站给出了 R 语言使用的分类方式。
- 数据库有更详细的数据分类方式，其中考虑了精度级别、固定长度或可变长度字段等因素。参见 W3Schools 的 SQL 指南。

1.2　矩形数据

矩形数据对象是数据科学分析中的典型引用结构，矩形数据对象包括电子表格、数据库表等。

主要术语

数据框

电子表格等矩形数据是统计和机器学习模型中的基本数据结构。

数据特征

通常称数据表中的一列为一个**特征**。

同义词：属性、输入、预测因子、变量

结果

不少数据科学项目涉及对**结果**的预测，常见的结果为“是”或“否”（例如表 1-1 中的“拍卖是否竞价？”）。**特征**有时在实验或研究中用于预测**结果**。

同义词：因变量、响应、目标、输出

记录

通常称数据表中的一行为一条**记录**。

同义词：事例、例子、实例、观察、模式、样本

矩形数据本质上是一个二维矩阵，其中行表示记录（事例），列表示特征（变量）。数据通常并非一开始就是矩阵形式的。例如，文本等非结构化数据必须先经处理和操作，才能表示为矩形数据形式的一系列特征（参见 1.1 节）。对于很多数据分析和建模任务，存储在关系型数据库中的数据必须先被抽取出来，并置于一张表中。

表 1-1 显示了测量数据或计数数据（ 例如“持续时间”和“成交价”）及分类数据（例如“分类”和“货币”）。如上所述，二元变量（例如“是”或“否”，0 或 1）是一种特殊的分类数据。表 1-1 的最右一列是一个指标变量，表示拍卖是否进行了竞价。

表1-1：一种常见的数据格式

分类	货币	卖家评级	持续时间	终止日期	成交价	开拍价	是否竞价
音乐 / 电影 / 游戏	美元	3249	5	周一	0.01	0.01	0
音乐 / 电影 / 游戏	美元	3249	5	周一	0.01	0.01	0

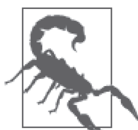
（续）

分类	货币	卖家评级	持续时间	终止日期	成交价	开拍价	是否竞价
汽车	美元	3115	7	周二	0.01	0.01	0
汽车	美元	3115	7	周二	0.01	0.01	0
汽车	美元	3115	7	周二	0.01	0.01	0
汽车	美元	3115	7	周二	0.01	0.01	0
汽车	美元	3115	7	周二	0.01	0.01	1
汽车	美元	3115	7	周二	0.01	0.01	1

1.2.1 数据框和索引

传统数据库表会指定一列或多个列为索引。索引可以极大地提高某些 SQL 查询的效率。在带有 pandas 数据分析库的 Python 中，基本的矩形数据结构是 `DataFrame` 对象，并且在默认情况下，Python 会根据 `DataFrame` 对象中行的次序，自动建立一个整数索引。pandas 数据分析库支持设置多级或层次索引，以提高特定操作的效率。

在 R 语言中，基本的矩形数据结构是 `data.frame` 对象。`data.frame` 隐含有基于行次序的整数索引。虽然用户可以使用 `row.names` 属性创建自定义键值，但是 R 语言的原生 `data.frame` 并不支持自定义索引或多级索引。`data.table` 和 `dplyr` 这两个新的 R 包解决了这一缺陷，因而得到了广泛的使用。它们都支持多级索引，可以显著提高 `data.frame` 的使用效率。



术语上的差异

矩形数据的术语可能令人困惑。对于同一事物，统计学家和数据科学家使用了不同的术语。统计学家在模型中使用**预测变量**去预测一个**响应**或**因变量**，而数据科学家使用**特征**去预测**目标**。还有一个同义词尤其令人困惑。对于一行数据，计算机科学家使用**样本**这一术语；而对于统计学家，一个**样本**意味着一个行的集合。

1.2.2 非矩形数据结构

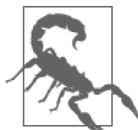
除了矩形数据之外，还有一些其他类型的数据。

时序数据记录了对同一变量的连续测量值。它是统计预测方法的原始输入数据，也是物联网设备所生成的数据的关键组成部分。

空间数据结构用于地图和定位分析，它比矩形数据结构更为复杂和多变。在**对象**表示中，空间数据关注的是对象（例如一所房子）及其空间坐标。与之形成对比的是，**字段**视图关注空间中的小单元及相关的度量值（例如像素点的亮度）。

图形（或网络）数据结构用于表示物理上的、社交网络上的和抽象的关系。例如，Facebook 或 LinkedIn 等社交网络图表示了人们在网络上的相互联系；由道路连接在一起的分布汇聚点构成了一个物理网络的例子。图形结构对于某些类型的问题十分有用，例如网络优化和推荐系统。

在数据科学中，每种数据类型都有其独特的方法论。本书关注的是矩形数据，它是预测建模的基本构件。



统计学中的图形

在计算机科学和信息技术中，**图形**通常指对实体间关联情况的描述及底层的数据结构。在统计学中，**图形**用于指代各种绘图和**可视化结果**，而不仅仅是指实体间的关联情况。这一术语只用于指代可视化，而非数据结构。

本节要点

- 矩阵是数据科学中的基本数据结构。在矩阵中，行是记录，列是变量（特征）。
- 术语中会存在一些令人困惑之处。在与数据科学相关的各学科中，例如统计学、计算机科学和信息技术等，存在着一系列的同义词。

1.2.3　拓展阅读

- R 语言中数据框的相关文档。
- Python 数据框的相关文档。

1.3　位置估计

变量表示了测量数据或计数数据，一个变量的取值可能会数以千计。探索数据的一个基本步骤，就是获得每个特征（变量）的“典型值”。典型值是对数据最常出现位置的估计，即数据的集中趋势。

主要术语

均值

所有数据值之和除以数值的个数。

同义词：平均值

加权均值

各数值乘以相应的权重值，然后加总求和，再除以权重的总和。

同义词：加权平均值

中位数

使得数据集中分别有一半数据位于该值之上和之下。

同义词：第 50 百分位数

加权中位数

使得排序数据集中分别有一半的权重之和位于该值之上和之下。

切尾均值

在数据集剔除一定数量的极值后，再求均值。

同义词：截尾均值

稳健

对极值不敏感。

同义词：耐抗性

离群值

与大部分数据值差异很大的数据值。

同义词：极值

乍一看，总结数据是一件十分简单的事情，对数据取**均值**即可（参见 1.3.1 节）。事实上，虽然均值易于计算，也便于使用，但在一般情况下，均值并非是对中心值的最好度量。因此，统计学家研究并提出了一些估计量，用于替代均值。



度量和估计量

统计学家通常使用**估计量**（estimate）一词表示从手头已有数据计算得到的值，用于描述所看到的数据情况与确切的（或理论上为真的）状态之间的差异。数据科学家和商业分析师更倾向于称这些由计算得到的值为**度量**（metric）。这一术语上的差异，反映了统计学家和数据科学家在方法上的不同。统计学的核心在于如何解释不确定度，而数据科学则关注如何解决一个具体的商业或企业目标。因此，统计学家使用估计量，而数据科学家使用度量。

1.3.1　均值

均值，又称**平均值**，是最基本的位置估计量。均值等于所有值的总和除以值的个数。例如，集合 {3, 5, 1, 2} 的均值是 (3 + 5 + 1 + 2)/4 = 11/4 = 2.75。一般使用符号 $\overline{x}$（读作“x 拔”）表示总体中一个样本的均值。给定 n 个数据值：$x_1, x_2, \cdots, x_N$，均值的计算公式为：

$$均值 = \overline{x} = \frac{\sum_i^n x_i}{n}$$



通常使用 N（或者 n）表示记录值或观测值的总数。在统计学中，如果我们指的是总体，则使用大写字母 N；如果指的是总体中的一个样本，则使用小写字母 n。但是在数据科学中，这一区别无关紧要，因此两种表示方式均可。

切尾均值是均值的一个变体。计算切尾均值时，需要在一个有序数据集的两端上去除一定数量的值，再计算剩余数值的均值。如果使用 $x_{(1)}, x_{(2)}, \cdots, x_{(n)}$ 表示一个有序数据集，其中 $x_{(1)}$ 是最小值，$x_{(n)}$ 是最大值，那么去除 p 个最大值和 p 个最小值的切尾均值的计算公式为：

$$\text{切尾均值} = \bar{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n-2p}$$

切尾均值消除了极值对均值的影响。举个例子，在国际跳水比赛中，会有五名裁判打分，一名选手的最终得分需要去除其中的最高分和最低分，取余下三名裁判打分的均值[2]。这确保了裁判难以操纵选手的得分，因为每名裁判可能会偏向自己国家的选手。切尾均值得到了广泛使用。相对于普通的均值，人们在很多情况下更倾向于使用切尾均值。1.3.2 节将对此做出详细的介绍。

另一种均值是**加权均值**。在计算加权均值时，要将每个值 x_i 乘以一个权重值 w_i，并将加权值的总和除以权重的总和。计算公式为：

$$\text{加权均值} = \bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i}^{n} w_i}$$

使用加权均值，主要是出于以下两个方面的考虑。

- 一些值本质上要比其他的值更为多变，因此需要对多变的观测值赋予较低的权重。例如，如果我们需要对来自多个传感器的数据计算均值，但是其中一个传感器的数据不是很准确，那么我们可对该传感器的数据赋予较低的权重。
- 所采集的数据可能并未准确地表示我们想要测量的不同群组。例如，受限于在线实验的开展方式，我们得到的数据集可能并未准确地反映出不同用户群组的情况。为了修正这一问题，我们可对未准确表示的群组赋予较高的权重。

1.3.2　中位数和稳健估计量

中位数是位于有序数据集中间位置处的数值。如果数值的个数为偶数，那么中位数实际上是位于中间位置处的两个值的均值。不同于使用所有观测值计算得到的均值，中位数仅取决于有序数据集中间位置处的值。尽管看上去中位数的计算方法存在一些弊端，但是考虑到均值对数据更敏感，因此在不少实际应用中，中位数依然是更好的位置度量。例如，我们想要了解西雅图华盛顿湖周边地区普通家庭的收入情况。在比较麦地那地区和温德米尔地区时，使用均值会产生迥异的结果，因为比尔·盖茨就生活在麦地那地区。如果使用中位数，那么统计结果完全不会受比尔·盖茨的影响。处于中间位置的观测值不会有变化。

有时我们需要计算**加权中位数**，这与使用加权均值的原因相同。和计算中位数一样，我们首先不考虑每个数值所关联的权重，对数据集排序。加权中位数并不是取有序数据集中间位置处的值，而是取可以使有序数据集上下两部分的权重总和相同的值。和中位数一样，加权中位数也对离群值不敏感。

离群值

我们称中位数为一种对位置的**稳健**估计量，因为它不会受**离群值**（极端情况）的影响，而离群值会使结果产生偏差。**离群值**是距离数据集中其他所有值都很远的值。尽管在各种数

注 2：“Diving.”Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 10 Mar 2016. Web. 19 Mar 2016.

据汇总和绘图中对离群值给出了一些惯例表示（参见 1.5.1 节），但是对离群值的准确定义还是摆脱不了主观性。离群值本身并不一定是无效的或错误的数据（如上例中比尔·盖茨的收入），但往往是由数据的错误所导致的，例如，混淆了数据的计量单位（混用了千米与米），或是传感器的读数不准确。如果离群值是由不准确的数据导致的，这时使用均值就会给出糟糕的位置估计，而使用中位数的估计则依然有效。无论是哪种情况，我们都应找出离群值，而且它们通常也值得进一步研究。



异常检测

在普通的数据分析中，离群值有时富含信息，有时则令人头疼。而**异常检测**则不同，它所关注的就是离群值，而其余大部分数据则用于定义“正常”的情况，即如何测定异常。

中位数并非唯一的稳健位置估计量。事实上，为了消除离群值的影响，也广泛地使用了切尾均值。例如，除非数据集的规模很小，否则通常我们会将数据集的开头和结尾各舍弃10%，以使数据集免受离群值的影响。切尾均值可以看作一种在中位数和均值之间的折中方案。它对数据集中的极值非常稳健，同时在计算位置估计量时使用了更多的数据。



其他稳健的位置估计量

统计学家还提出了很多其他的位置估计量，主要目的在于提供比均值更稳健、更高效的估计量。其中更高效是指，能够更好地分辨数据集中的微小位置差异。这些估计量通常适用于小规模的数据集，而对于大规模乃至中等规模的数据集，它们并不能提供更多的帮助。

1.3.3　位置估计的例子：人口和谋杀率

表 1-2 显示了一个数据集的前几行数据，其中包含了美国各州的人口数量和谋杀率，单位为每年每十万人中被谋杀的人数。

表1-2：`data.frame`中的几行数据，列出了美国各州的人口数量和谋杀率

	州	人口	谋杀率
1	阿拉巴马州	4 779 736	5.7
2	阿拉斯加州	710 231	5.6
3	亚利桑那州	6 392 017	4.7
4	阿肯色州	2 915 918	5.6
5	加利福尼亚州	37 253 956	4.4
6	科罗拉多州	5 029 196	2.8
7	康涅狄格州	3 574 097	2.4
8	特拉华州	897 934	5.8

下面使用 R 语言计算美国各州人口的均值、切尾均值和中位数。

```
> state <- read.csv(file="/Users/andrewbruce1/book/state.csv")
> mean(state[["Population"]])
[1] 6162876
> mean(state[["Population"]], trim=0.1)
[1] 4783697
> median(state[["Population"]])
[1] 4436370
```

我们看到，均值大于切尾均值，而切尾均值大于中位数。

这是因为，切尾均值分别去除了最大的和最小的五个州（`trim=0.1` 表示在最大端和最小端分别去除 10% 的数据）。如果要计算美国的平均谋杀率，那么需要使用加权均值或中位数，这两个度量考虑了各州的人口差异。R 语言并未提供计算加权中位数的函数，因此我们需要安装额外的软件包，比如 `matrixStats`。代码如下。

```
> weighted.mean(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.445834
> library("matrixStats")
> weightedMedian(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.4
```

在本例中，加权中位数和中位数大体相同。

本节要点

- 均值是一种基本的位置度量，但是对极值（离群值）敏感。
- 其他一些度量更为稳健，例如中位数和切尾均值。

1.3.4 拓展阅读

- 对于计算基本的位置度量，美国普度大学的 Michael Levine 提供了一些有用的课程讲义。
- 约翰・图基的经典著作 *Exploratory Data Analysis* 至今依然广为阅读。

1.4 变异性估计

位置只是总结特性的一个维度，另一个维度是**变异性**（variability），也称**离差**（dispersion），它测量了数据值是紧密聚集的还是发散的。变异性是统计学的一个核心概念，统计学关注如何测量变异性，如何降低变异性，如何识别真实变异性中的随机性，如何识别真实变异性的各种来源，以及如何在存在变异性的情况下做出决策。

主要术语

偏差

位置的观测值与估计值间的直接差异。

同义词：误差、残差

方差

对于 n 个数据值，方差是对距离均值的偏差平方后求和，再除以 n−1。

同义词：均方误差

标准偏差

方差的平方根。

同义词：L2 范数、欧几里得范数

平均绝对偏差

对数据值与均值间偏差的绝对值计算均值。

同义词：L1 范数、曼哈顿范数

中位数绝对偏差

数据值与中位数间绝对偏差的均值。

极差

数据集中最大值和最小值间的差值。

顺序统计量

基于从大到小排序的数据值的度量。

同义词：秩

百分位数

表示一个数据集中，P% 的值小于或等于第 P 百分位数，(100−P)% 的值大于或等于第 P 百分位数。

同义词：四分位数

四分位距

第 75 百分位数和第 25 百分位数间的差值。

同义词：四分位差

正如对位置有均值、中位数等多种不同的测定方式，变异性也有多种不同的测定方式。

1.4.1 标准偏差及相关估计值

使用最广泛的变异性估计量基于位置估计值和观测数据值间的差异或**偏差**。给定一个数据集 {1, 4, 4}，其均值是 3，中位数是 4。各个数据与均值的偏差分别为：1−3 = −2，4−3 = 1，4−3 = 1。这些偏差值说明了数据围绕中心值的分散程度。

一种测量变异性的方法是，估计这些偏差的一个典型值。然而，对这些偏差值本身取均值是无法给出更多信息的，因为负的偏差值将会抵消正的偏差值。事实上，相对于均值的偏差值的总和为零。一种简单的方法是对均值偏差的绝对值取均值。在上例中，各偏差的绝

对值分别是 2、1 和 1，它们的均值为 (2 + 1 + 1)/3 = 1.33。这就是**平均绝对偏差**，计算公式为：

$$平均绝对偏差 = \frac{\sum_{i=1}^{n} | x_i - \bar{x} |}{n}$$

其中 $\bar{x}$ 是样本的均值。

最广为人知的变异性估计量是**方差**和**标准偏差**，它们基于偏差的平方。方差是偏差平方值的均值，而标准偏差是方差的平方根。

$$方差 = s^2 = \frac{\sum (x - \bar{x})^2}{n-1}$$

$$标准偏差 = s = \sqrt{\frac{\sum (x - \bar{x})^2}{n-1}}$$

标准偏差比方差更易于理解，因为它具有与原始数据相同的尺度。然而，考虑到标准偏差的计算公式相对更复杂，并且不太直观，大家可能会奇怪，为什么统计学中更愿意使用标准偏差，而不是平均绝对偏差。这是由于标准偏差在统计学理论中的领导地位。从数学角度来看，使用平方值要比使用绝对值更方便，尤其是对于统计模型。

自由度是 *n*，还是 *n*–1？

在统计学书籍中，总是存在这样一个讨论：计算方差时的被除数为什么是 *n*–1，而不是 *n*？这一讨论引出了**自由度**的概念。计算结果的差别并不大，这是因为通常 *n* 总是足够大，以至于除以 *n* 或除以 *n*–1 时，结果并不会有很大的差别。如果你关注这一问题，我们在此解释一下原因。这是基于你想要根据样本估计总体这一前提。

如果在方差公式中使用了直观的除数 *n*，那么就会低估方差的真实值和总体的标准偏差。这被称为**有偏估计**。但是，如果除以 *n*–1 而不是 *n*，这时标准偏差就是**无偏估计**。

要完整地解释为什么使用 *n* 会导致有偏估计，这就涉及自由度的概念。自由度考虑了计算估计量中的限制个数。在这种情况下，自由度是 *n*–1，因为其中有一个限制：标准偏差依赖于计算样本的均值。对于很多问题而言，数据科学家并不需要担心自由度的问题。但是在某些情况下，自由度十分重要（参见 6.1.5 节）。

无论方差、标准偏差，还是平均绝对偏差，它们对离群值和极值都是不稳健的（参见 1.3.2 节）。其中，方差和标准偏差对离群值尤为敏感，因为它们基于偏差的平方值。

中位数绝对偏差（MAD）是一种稳健的变异性估计量。中位数绝对偏差的计算公式为：

$$\text{MAD} = 中位数(| x_1 - m |, | x_2 - m |, \cdots, | x_n - m |)$$

其中，m 是中位数。和中位数一样，中位数绝对偏差也不受极值的影响。我们可以参考切尾均值的计算方法（参见 1.3.1 节），计算切尾标准偏差。



即使数据符合正态分布，方差、标准偏差、平均绝对偏差以及中位数绝对偏差这四者也并非是等价的估计量。事实上，标准偏差总是大于平均绝对偏差，而平均绝对偏差总是大于中位数绝对偏差。有时，中位数绝对偏差会乘上一个常数比例因子（通常使用 1.4826），使得在正态分布下，中位数绝对偏差与标准偏差具有相同的尺度。

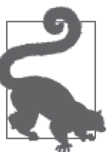
1.4.2 基于百分位数的估计量

另一种估计离差的方法基于对有序数据分布情况的查看。基于有序数据的统计量被称为**顺序统计量**，其中最基本的测量是**极差**，即数据的最大值与最小值之间的差值。知道最大值和最小值本身也是十分有用的，这有助于识别离群值。但是极差对离群值非常敏感，对于测量数据的离差并非十分有用。

为避免对离群值敏感，我们可以删除有序数据两端的值，然后再查看数据的极差。正式表述为，此估计量基于**百分位数**间的差异。在一个数据集中，第 P 百分位数表明，至少有 P% 的值小于或等于该值，而 $(100-P)$% 的值大于或等于该值。例如，如果要找到第 80 百分位数，我们首先对数据进行排序，然后从最小值开始，按照从小到大的顺序数出其中 80% 的数值。注意，中位数等同于第 50 百分位数。百分位数在本质上等同于**四分位数**，而四分位数是根据分数做索引的，因此 0.8 四分位数等同于第 80 百分位数。

变异性的一种常用测量方法第 25 百分位数和第 75 百分位数间的差值，称为**四分位距**（IQR）。下面给出一个例子，对于数据集 {3, 1, 5, 3, 6, 7, 2, 9}，我们在排序后得到 {1, 2, 3, 3, 5, 6, 7, 9}。其中第 25 百分位数是 2.5，第 75 百分位数是 6.5，因此四分位距就是 6.5−2.5 = 4。不同软件在计算方法上可能会稍有差异，并给出不同的答案（参见本节末给出的知识点），但是差异通常很小。

对于规模非常大的数据集，准确计算百分位数的成本很高，因为需要对所有的数据做排序。在机器学习和统计软件中，使用了一些特殊的算法[3]，这些算法可以快速计算出一个近似的百分位数，并有一定的准确度。



四分位距的准确定义

如果一个数据集中的数值个数是偶数（即 n 是偶数），那么根据上面的定义，百分位数不是唯一的。事实上，我们可以取任意一个位于顺序统计量 $x_{(j)}$ 和 $x_{(j+1)}$ 间的值，只要 j 满足：

$$100\times\frac{j}{n}\leqslant P<100\times\frac{j+1}{n}$$

正式的表述是，百分位数是一种加权平均：

$$\text{百分位数 }(P)=(1-w)x_{(j)}+wx_{(j+1)}$$

其中，权重值 w 介于 0 和 1 之间。不同的统计软件选取 w 的方法略有不同。R 语言的 `quantile` 函数提供了 9 种计算百分位数的方法。除非数据集的规模很小，否则通常我们不需要操心百分位数的准确计算方法。

注 3：Zhang, Qi and Wang, Wei. 19th International Conference on Scientific and Statistical Database Management, IEEE Computer Society (2007).

1.4.3　例子：美国各州人口的变异性估计量

本例的数据集中包括了美国各州的人口数量和谋杀率。表 1-3 显示了该数据集的前几行数据。

表1-3：`data.frame`的几行数据，显示了按州统计的人口数量和谋杀率

	州	人口	谋杀率
1	阿拉巴马州	4 779 736	5.7
2	阿拉斯加州	710 231	5.6
3	亚利桑那州	6 392 017	4.7
4	阿肯色州	2 915 918	5.6
5	加利福尼亚州	37 253 956	4.4
6	科罗拉多州	5 029 196	2.8
7	康涅狄格州	3 574 097	2.4
8	特拉华州	897 934	5.8

下面使用 R 语言自带的函数计算标准偏差、四分位距和中位数绝对偏差。通过下面的计算，我们得到了美国各州人口数据的变异性估计量。

```
> sd(state[["Population"]])
[1] 6848235
> IQR(state[["Population"]])
[1] 4847308
> mad(state[["Population"]])
[1] 3849870
```

可以看到，标准偏差几乎是中位数绝对偏差的两倍（在 R 语言中，默认将中位数绝对偏差的规模调整到与均值一样）。这并不奇怪，因为标准偏差对离群值敏感。

本节要点

- 方差和标准偏差是日常最广为使用的变异性统计量。
- 方差和标准偏差都对离群值敏感。
- 更稳健的度量包括偏离均值（百分位数、四分位距）的平均（中位数）绝对偏差。

1.4.4　拓展阅读

- David Lane 的在线统计学资源中有一节介绍了百分位数。
- Kevin Davenport 在 R-Bloggers 上撰写了一篇有用的文章，介绍了距离中位数的各种偏差以及它们的稳健性。

1.5　探索数据分布

前文介绍的各种估计量都是通过将数据总结为单一数值，去描述数据的位置或变异性。这些估计量可用于探索数据的整体分布情况。

主要术语

箱线图

图基提出的一种绘图，是一种快速可视化数据分布情况的方法。

同义词：箱形图、箱须图

频数表

将数值型数据的计数情况置于一组间隔（组距）中。

直方图

对频数表的绘图，其中 x 轴是组距，y 轴是计数（或比例）。

密度图

直方图的平滑表示，通常基于某种**核密度估计**。

1.5.1 百分位数和箱线图

在 1.4.2 节中，我们介绍了如何使用百分位数测量数据的分布情况。百分位数对于总结整体分布也十分有用。报告中经常会用到**四分位数**（即第 25 百分位数、第 50 百分位数和第 75 百分位数）和**十分位数**（即第 10 百分位数、第 20 百分位数……第 90 百分位数）。在总结数据**尾部**情况（外延范围）时，百分位数尤为有用。在大众文化中，也有**百分之一阵营**（one-percenter）的说法，它指的是拥有 99% 的财富的富人。

表 1-4 中列出了美国一些州的谋杀率的百分位数。在 R 语言中，可使用 `quantile` 函数生成百分位数。

```
quantile(state[["Murder.Rate"]], p=c(.05, .25, .5, .75, .95))
   5%   25%   50%   75%   95%
1.600 2.425 4.000 5.550 6.510
```

表1-4：按州谋杀率的百分位数

5%	25%	50%	75%	95%
1.60	2.42	4.00	5.55	6.51

上例中，中位数为每十万人中有四名谋杀犯，但是其中存在一些变异性。第 5 百分位数是 1.60，第 95 百分位数是 6.51。

箱线图是由图基提出的一种快速可视化绘图[4]，它基于百分位数可视化数据的分布。图 1-2 显示了按州划分人口的箱线图，它是由下面的 R 命令生成的。

```
boxplot(state[["Population"]]/1000000, ylab="Population (millions)")
```

注 4：Tukey, John W. *Exploratory Data Analysis*. Pearson (1977). ISBN:978-0-201-07616-5.

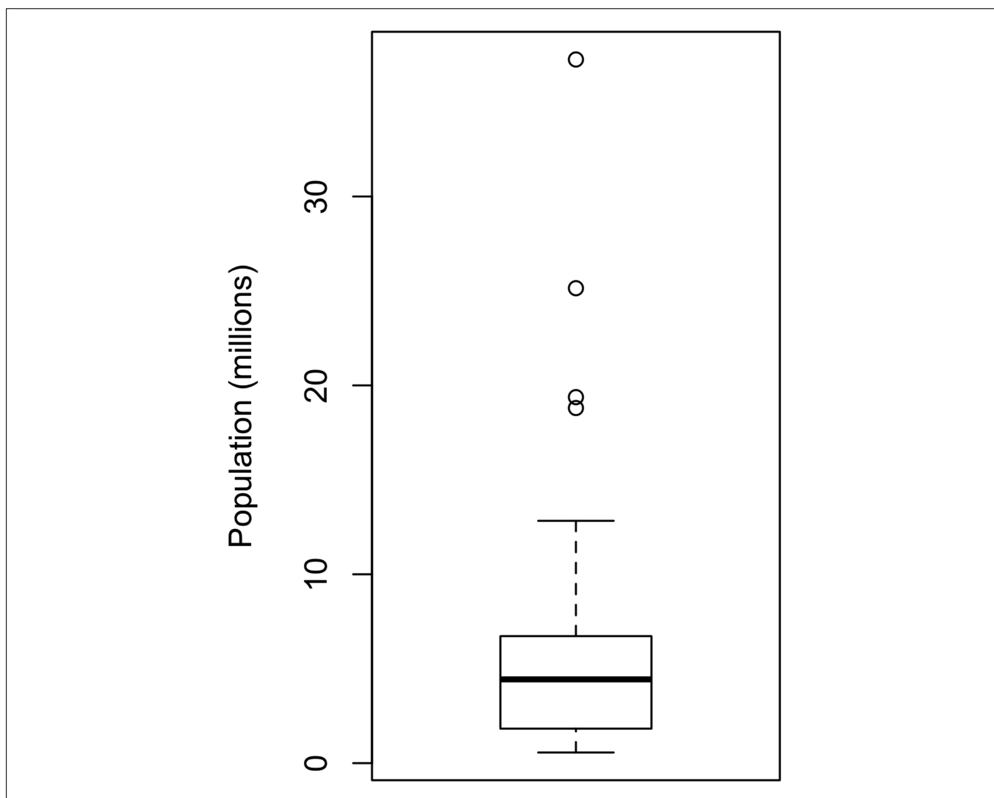


图 1-2：美国各州人口的箱线图

箱子的顶部和底部分别是第 75 百分位数和第 25 百分位数。箱内的水平线表示的是中位数。图中的虚线称为**须**（whisker）。须从最大值一直延伸到最小值，显示了数据的极差。箱线图有多种变体，具体细节可参考“R 文档：`boxplot` 函数”[5] 等资料。在默认情况下，该 R 函数会扩展须到箱子外的最远点，但不会超过四分位距的 1.5 倍。其他软件可能会采用不同的规则。在须外的所有数据绘制为单个点。

1.5.2　频数表和直方图

变量的频数表将该变量的极差均匀地分割为多个等距分段，并给出落在每个分段中的数值个数。在 R 语言中，可使用下面命令计算美国人口按州分布的频数表，结果显示在表 1-5 中。

```
breaks <- seq(from=min(state[["Population"]]),
              to=max(state[["Population"]]), length=11)
pop_freq <- cut(state[["Population"]], breaks=breaks,
              right=TRUE, include.lowest = TRUE)
table(pop_freq)
```

注 5：R Core Team.“R: A Language and Environment for Statistical Computing,”R Foundation for Statistical Computing (2015).

表1-5：美国人口按州分布的频数表

组距编号	组距范围	计数	州名
1	563 626 ～ 4 232 658	24	怀俄明州、佛蒙特州、北达科他州、阿拉斯加州、南达科他州、特拉华州、蒙大拿州、罗得岛州、新罕布什尔州、缅因州、夏威夷州、爱达荷州、内布拉斯加州、西弗吉尼亚州、新墨西哥州、内华达州、犹他州、堪萨斯州、阿肯色州、密西西比州、爱荷华州、康涅狄格州、俄克拉荷马州、俄勒冈州
2	4 232 659 ～ 7 901 691	14	肯塔基州、路易斯安那州、南卡罗来纳州、阿拉巴马州、科罗拉多州、明尼苏达州、威斯康辛州、马里兰州、密苏里州、田纳西州、亚利桑那州、印第安纳州、马萨诸塞州、华盛顿州
3	7 901 692 ～ 11 570 724	6	弗吉尼亚州、新泽西州、北卡罗来纳州、乔治亚州、密歇根州、俄亥俄州
4	11 570 725 ～ 15 239 757	2	宾夕法尼亚州、伊利诺伊州
5	15 239 758 ～ 18 908 790	1	佛罗里达州
6	18 908 791 ～ 22 577 823	1	纽约州
7	22 577 824 ～ 26 246 856	1	得克萨斯州
8	26 246 857 ～ 29 915 889	0	
9	29 915 890 ～ 33 584 922	0	
10	33 584 923 ～ 37 253 956	1	加利福尼亚州

根据 2010 年的人口普查，美国人口最少的州是怀俄明州，563 626 人；人口最多的州是加利福尼亚州，37 253 956 人。极差为 37 253 956−563 626 = 36 690 330。我们必须将极差划分为大小相等的组距，假定为 10 个组距。这样，每个组距的宽度为 3 669 033。第一个组距的范围是从 563 626 到 4 232 658。最后一个组距的范围是从 33 584 923 到 37 253 956，其中只有加利福尼亚一个州。加利福尼亚州之前的两个组距是空的，直到得克萨斯州。添加空组距也是有必要的；空组距中没有值，这一事实是很有价值的信息。尝试不同大小的组距也是非常有用的。如果组距过大，那么就会隐藏掉分布的一些重要特性。如果组距过小，那么结果就会过于颗粒化，失去查看整体图的能力。



频数表和百分位数都是通过创建组距总结数据。一般情况下，四分位数和十分位数在每个组距中具有相同的计数，但是每个组距的大小不同，我们称之为等计数组距。与之相对，如果频数表在每个组距中的计数不同，我们称之为等规模组距。

直方图是频数表的一种可视化方法，其中 x 轴为组距，y 轴为数据的计数。在 R 语言中，要创建对应于表 1-5 的直方图，可使用指定了 `breaks` 参数的 `hist` 函数，命令如下。

```
hist(state[["Population"]], breaks=breaks)
```

图 1-3 显示了上面命令生成的直方图。一般说来，在绘制直方图时应注意以下几点。

- 空组距也应包括在直方图中。
- 各组距是等宽的。
- 组距的数量（或组距的大小）取决于用户。
- 各条块相互紧挨着，条块间没有任何空隙，除非存在空组距。

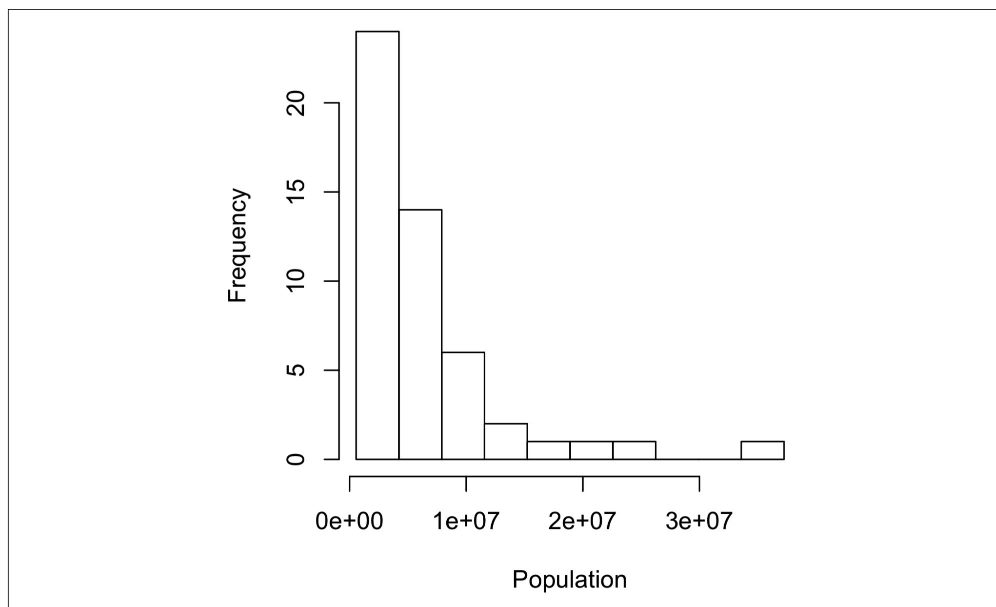
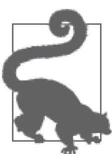


图 1-3：美国各州人口的直方图



统计学中的矩（moment）

在统计学理论中，位置和变异性分别称为分布的**一阶矩**和**二阶矩**，而分布的三阶矩和四阶矩分别被称为**偏度**和**峰度**。偏度显示了数据是偏向较小的值还是偏向较大的值，峰度则显示了数据中具有极值的倾向性。通常情况下，我们并不使用度量去测定偏度和峰度，而是通过图 1-2 和图 1-3 这样的可视化方法发现它们。

1.5.3 密度估计

密度图与直方图有关，它用一条连续的线显示数据值的分布情况。我们可以将密度图看作由直方图平滑得到的，尽管它通常是使用一种**核密度估计量**[6]从数据中直接计算得到的。图 1-4 将密度估计情况显示在直方图上。在 R 语言中，可以使用 `density` 函数计算密度估计。

```
hist(state[["Murder.Rate"]], freq=FALSE)
lines(density(state[["Murder.Rate"]]), lwd=3, col="blue")
```

注 6：Duang, Tarn.“An introduction to kernel density estimation”(2001).

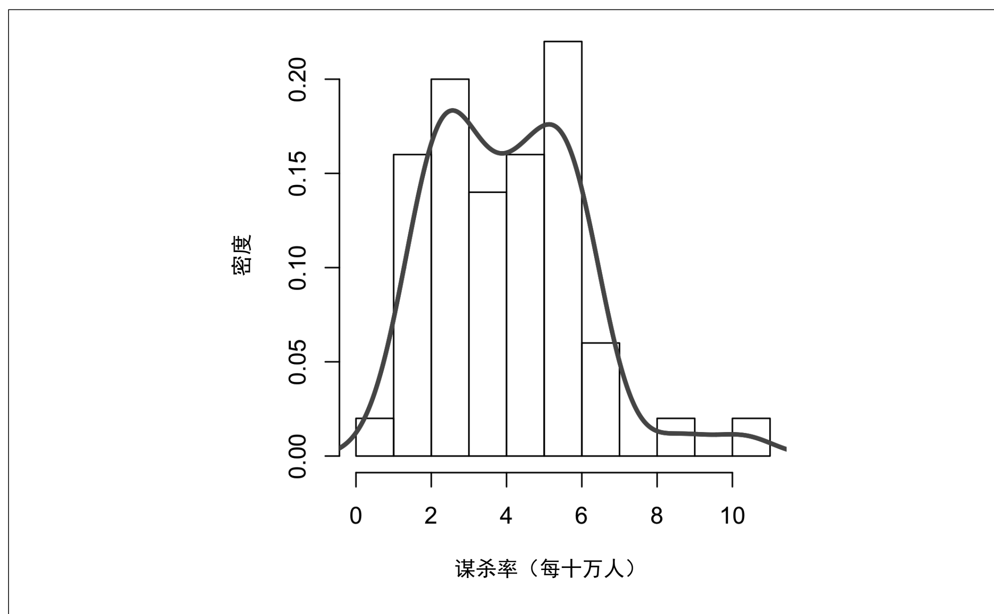


图 1-4：美国各州谋杀率的密度图

与图 1-3 中的直方图相比，图 1-4 的不同之处在于 y 轴的尺度。密度图相当于按比例而非按计数绘制直方图。因此在 R 语言中，我们需要指定参数 `freq=FALSE`。



密度估计

密度估计是一个很宽泛的话题，在统计学研究中具有悠久的历史。事实上，已有二十多个 R 包提供了计算密度估计的函数。Henry Deng 和 Hadley Wickham 在“Density estimation in R”[7]一文中对 R 包进行了综述，其中特别推荐了 ASH 和 KernSmooth。对于许多数据科学问题，完全不必操心各种类型的密度估计，R 语言的基本函数就完全够用了。

本节要点

- 频数直方图在 y 轴上绘制频数计数，在 x 轴上绘制变量值。它提供了对数据分布的概览。
- 频数表是直方图中频数计数的表格形式。
- 在箱线图中，箱子的顶部和底部分别表示第 75 百分位数和第 25 百分位数。箱线图也提供了数据分布的基本情况。多个箱线图通常是并排展示的，以便于比较分布情况。
- 密度图是直方图的一种平滑表示。它需要一个基于数据估计绘图的函数（当然也可以做多个估计）。

注 7：Deng, H. and Wickham, H.“Density estimation in R”(2011).

1.5.4　拓展阅读

- 美国纽约州立大学奥斯威戈分校的一位教授给出了创建箱线图的步骤。
- Henry Deng 和 Hadley Wickham 在其论文“Density estimation in R”中介绍了如何在 R 语言中实现密度估计。
- R-Bloggers 网站提供了一篇关于 R 语言中直方图的有用文章，其中介绍了如何选取箱子大小等元素。
- R-Bloggers 网站还提供了一些介绍 R 语言中箱线图的类似文章。

1.6　探索二元数据和分类数据

使用基本的比例或百分比，我们就能了解分类数据的情况。

主要术语

众数

数据集中出现次数最多的类别或值。

期望值

如果类别可以与一个数值相关联，可以根据类别的出现概率计算一个平均值。

条形图

在绘图中，以条形表示每个类别出现的频数或占比情况。

饼图

在绘图中，圆饼中的一个扇形部分表示每个类别出现的频数或占比情况。

总结二元变量的情况，或总结只有几个类别的分类变量，是非常容易实现的，我们只需计算出数据中 1 的比例，或是重要类别出现的比例。例如，表 1-6 给出了按延迟原因分类的航班延迟的百分比，数据来自于美国达拉斯沃斯堡机场自 2010 年以来的延迟数据。延迟原因可分类为：航空公司管理原因、流量控制（ATC）系统延误、天气原因、安全原因以及到港航班延迟。

表1-6：美国达拉斯沃斯堡机场的航班延迟百分比，按延迟原因分类

航空公司管理原因	ATC	天气原因	安全原因	到港航班延迟
23.02	30.40	4.03	0.12	42.43

条形图是在各大媒体上常用的一种可视化工具，它可显示单个分类变量的总体情况。在条形图中，x 轴列出类别，y 轴表示频数或比例。图 1-5 显示了美国达拉斯沃斯堡机场每年按延迟原因分类的航班延迟情况，它是使用 R 语言的函数 `barplot` 生成的。

```
barplot(as.matrix(dfw)/6, cex.axis=.5)
```

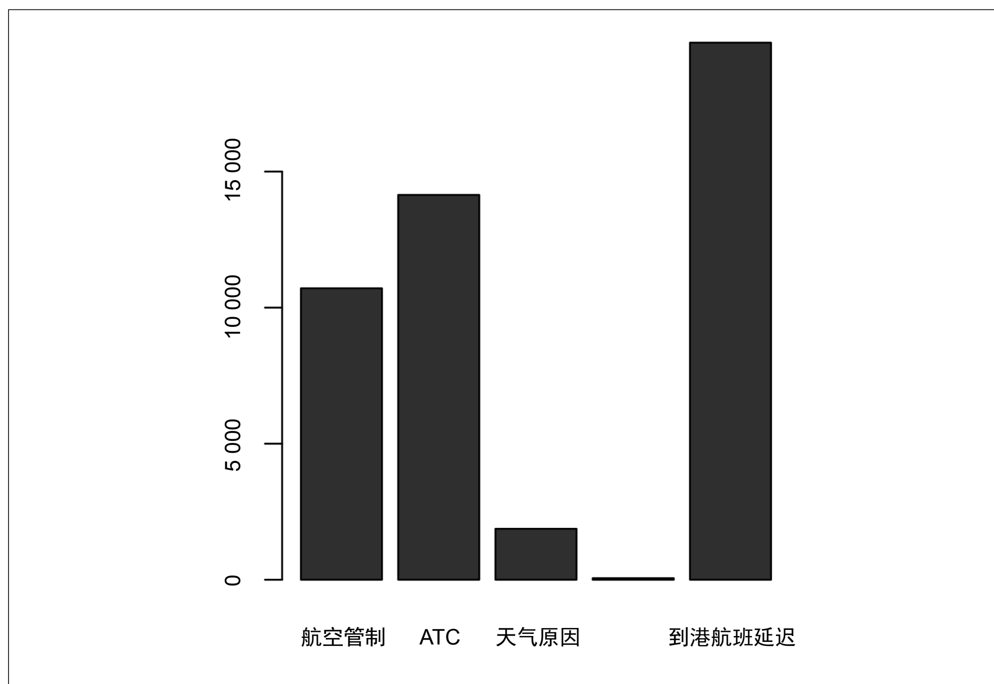


图 1-5：美国达拉斯沃斯堡机场航班延迟原因的条形图

注意，虽然条形图非常类似于直方图，但两者间存在一些差异。在条形图中，x 轴表示因子变量的不同类别，而在直方图中，x 轴以数值度量的形式表示某个变量的值。另外，在直方图中，通常各个条形是相互紧挨着显示的，条形间的间隔表示了数据中未出现的值；而在条形图中，各个条形的显示是相互独立的。

饼图是条形图的一种替代形式。统计学家和数据可视化专业人员通常不使用饼图。在他们看来，饼图在视觉上缺乏信息量[8]。



如何将数值型数据转换为分类数据

在 1.5.2 节中，我们通过对数据创建组距，生成了频数表。这同时也将数值型数据转换为排序的因子。就此而言，直方图和条形图是类似的，除非条形图中 x 轴上的类别不是有序的。将数值型数据转换为分类数据是非常重要的，它是数据分析中的一个重要步骤。该转换降低了数据的复杂度和规模，并有助于发现特征间的关系，尤其是在分析的初始阶段。

1.6.1　众数

众数指数据中出现最频繁的一个或一组数值。例如，美国达拉斯沃斯堡机场延迟原因的众

注 8：Few, Stephen. “Save the Pies for Dessert.” Visual Intelligence Newsletter, Perceptual Edge (2007).

数是“到港航班延迟”。再比如，基督教是美国绝大部分地区宗教信仰倾向的众数。众数是分类数据的一个基本汇总统计量，通常不用于数值型数据。

1.6.2 期望值

分类数据还有一个特殊类型，即数据类别可以表示成（或映射到）同一尺度的离散值。例如，新兴云技术的服务商提供了两种服务，一种服务的费用为每月 300 美元，另一种为每月 50 美元。服务商会举办免费的网络研讨会，以发现一些潜在的用户。来自企业的数据表明，有 5% 的研讨会参与者将会注册每月 300 美元的服务，15% 的参与者会注册每月 50 美元的服务，另外 80% 的人将不会注册任何服务。这样，我们可以将数据总结为一个**期望值**，估计企业的营业收入。期望值是一种加权均值，权重使用的是类别出现的概率。

期望值的计算方法如下。

(1) 输出值乘以其出现的概率。
(2) 将这些值加起来。

就上面给出的云服务例子而言，与会者支付服务费的期望值是每月 22.5 美元，计算过程如下。

$$\text{期望值} = 0.05 \times 300 + 0.15 \times 50 + 0.80 \times 0 = 22.5$$

期望值实际上是一种加权均值，其中加入了未来期望和概率权重的概念，所使用的概率通常是根据主观判断得到的。期望值是商业估值和资金预算中的一个基本概念，例如，一次新收购在未来 5 年中利润的期望值，或者一个诊所的新患者管理软件在节约开支上的期望值。

本节要点

- 分类数据通常按比例总结，可以使用条形图将它可视化。
- 类别用于表示不同类型的事物（例如苹果和橘子，男性和女性）、因子变量的等级（例如低、中和高），或由组距分隔的数值型数据。
- 期望值是对每个数值与该数值出现概率的乘积求和，通常用于总结因子变量的等级。

1.6.3 拓展阅读

只有了解误导性图，统计学课程才是完备的。误导性图通常涉及条形图和饼图。

1.7 相关性

无论是在数据科学还是研究中，很多建模项目的探索性数据分析都要检查预测因子之间的相关性，以及预测因子和目标变量之间的相关性。给定变量 X 和 Y，它们均有测量数据。如果变量 X 的高值随变量 Y 的高值的变化而变化，并且 X 的低值随 Y 的低值的变化而变化，那么我们称 X 和 Y 是正相关的。如果 X 的高值随 Y 的低值的变化而变化，反之亦然，那么我们称变量 X 和 Y 是负相关的。

主要术语

相关系数

一种用于测量数值变量间相关程度的度量，取值范围在 −1 到 +1 之间。

相关矩阵

将变量在一个表格中按行和列显示，表格中每个单元格的值是对应变量间的相关性。

散点图

在绘图中，*x* 轴显示一个变量的值，*y* 轴显示另一个变量的值。

考虑下面两个变量 v1 和 v2。它们是完全相关的，因为每个变量中的观测值都是按从小到大排列的。

v1: {1, 2, 3}
v2: {4, 5, 6}

向量点积是 4 + 10 + 18 = 32。现在我们尝试将其中一个变量中的观测值随机重排，然后重新计算二者的点积，所得到的值永远不会大于 32。因此，我们可以将向量点积作为一个度量，即做任意次随机排序后，向量点积都不会大于 32。（事实上，这一理念是与基于重抽样的估计密切相关的，参见 3.1 节。）尽管如此，该度量值并非很有意义，除非用于重抽样分布。

点积的一种标准化变体就是**相关系数**，该度量更为有用。对于两个总是保持同一尺度的变量，相关系数给出了两者间相关性的估计值。在计算**皮尔逊相关系数**时，要将变量 v1 的平均偏差乘以变量 v2 的平均偏差，再除以标准偏差之积，计算公式如下。

$$r = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{(n-1)s_x s_y}$$

注意，公式中使用的除数是 n−1，而不是 n。具体细节参见 1.4.1 节中关于“自由度是 n，还是 n−1？”的讨论。相关系数的值总是位于 +1（完全正相关）和 −1（完全负相关）之间。0 表示没有相关性。

变量的相关性可以是非线性的。在这种情况下，相关系数就不再是一种有用的度量。比如，税率和收入增加之间的关系。当税率由零开始增加时，收入也在增加。但是税率一旦达到一定高的水平并逼近 100% 时，这时避税增加了，而税收则实际下降了。

表 1-7 被称为**相关矩阵**，它显示了自 2012 年 7 月到 2015 年 6 月间的电信类股票每日收益间的相关性。从中可以看到，股票 Verizon（VZ）与 ATT（T）间的相关性最高，而与通信基础设施运营企业 Level Three（LVLT）间的相关性最低。需要注意的是，相关矩阵的对角线元素为 1（即一支股票与其自身的相关性是 1），并且对角线上下对称位置的信息是冗余的。

表1-7：电信类股票收益间的相关性

	T	CTL	FTR	VZ	LVLT
T	1.000	0.475	0.328	0.678	0.279
CTL	0.475	1.000	0.420	0.417	0.287
FTR	0.328	0.420	1.000	0.287	0.260
VZ	0.678	0.417	0.287	1.000	0.242
LVLT	0.279	0.287	0.260	0.242	1.000

通常，我们会用绘图展示表 1-7 这样的相关性表，实现对多个变量间关系的可视化。图 1-6 显示了主要 ETF（交易所交易资金）每日收益间的相关性。使用 R 语言中的 `corrplot` 软件包，很容易创建这样的绘图。命令如下。

```
etfs <- sp500_px[row.names(sp500_px)>"2012-07-01",
                 sp500_sym[sp500_sym$sector=="etf", 'symbol']]
library(corrplot)
corrplot(cor(etfs), method = "ellipse")
```

标准普尔 500 指数（SPY）和道琼斯指数（DIA）的 ETF 具有很高的相关性。类似地，主要由技术企业组成的 QQQ 和 XLK 指数是正相关的。保守 ETF，例如金价追踪指数（GLD）、原油价格指数（USO）或市场波动指数（VXX），倾向于与其他 ETF 负相关。椭圆的长轴方向显示了两个变量是正相关的还是负相关的：椭圆长轴方向偏右，为正相关；椭圆长轴方向偏左，为负相关。椭圆的阴影和宽度显示了关联的强度，更细长并且颜色更深的椭圆，对应于更强的相关性。

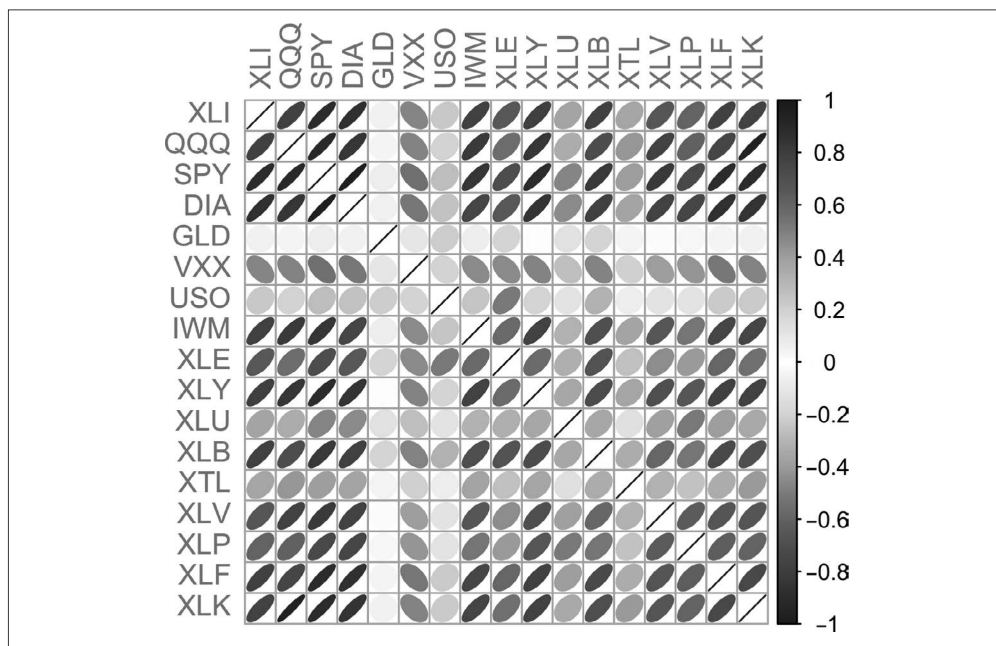


图 1-6：ETF 每日收益间的相关性

与平均值和标准偏差一样，相关系数同样对数据离群值敏感。对于经典的相关系数，有的软件包提供了一些稳健的替代方法。例如，R 语言的 `cor` 函数具有 `trim` 参数[9]，设置类似于截尾均值的计算[10]。



其他相关性估计量

统计学家早就提出了其他类型的相关系数，例如**斯皮尔曼秩相关系数**（Spearman's rho）、**肯德尔秩相关系数**（Kendall's tau）等基于数据秩的相关系数。由于这些估计量操作的是秩，而非数据值，所以它们对离群值稳健，并可以处理特定类型的非线性。但是在探索性数据分析中，数据科学家通常会坚持使用皮尔逊相关系数及其一些稳健的替代者。多数情况下，基于秩的估计量适用于小规模的数据集以及特定的假设检验。

1.7.1 散点图

散点图是一种可视化两个测量数据变量间关系的标准方法。在散点图中，x 轴表示一个变量，y 轴表示另一个变量，图中的每个点对应于一条记录。图 1-7 显示了股票 ATT 和 Verizon 日收益的绘图，用下面的 R 命令生成。

```
plot(telecom$T, telecom$VZ, xlab="T", ylab="VZ")
```

从图中可以看到，两支股票的日收益具有强正相关性。在大部分交易日中，两支股票都保持同步涨跌。但还有少数几个交易日，其中一支股票明显下跌而另一支股票上涨，或是相反。

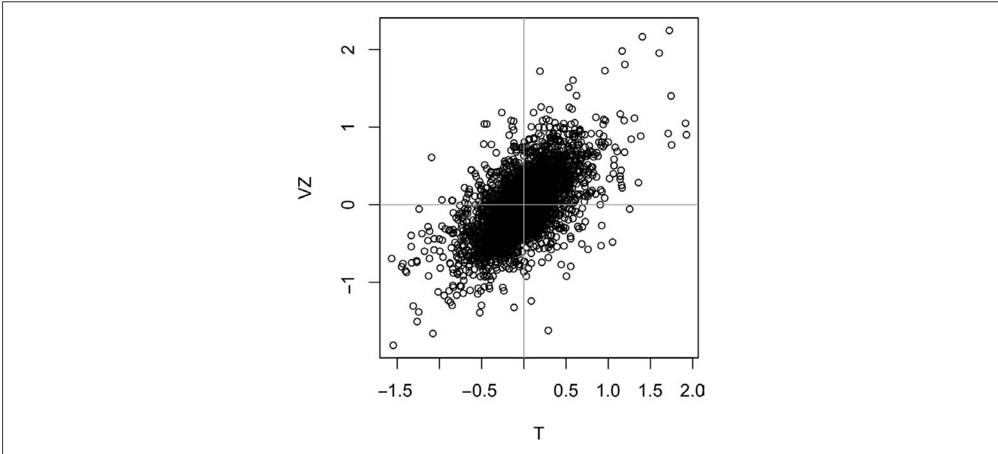


图 1-7：股票 ATT 和 Verizon 日收益的散点图

注 9：原文即是如此。事实上，R 的 `cor` 函数并不提供 `trim` 参数。在 R 中，有多种方法实现对离群值稳健的相关性计算。一种是通过设置 `cor` 函数的 `method` 参数，`method = "spearman"` 或 `method = "kendall"`，原因参见本节知识点“其他相关性估计量”。另一种方法是使用其他 R 软件包提供的函数，例如，`robust` 软件包的 `covRob` 函数、`MASS` 软件包的 `rlm` 函数等。——译者注

注 10：R Core Team.“R: A Language and Environment for Statistical Computing,”R Foundation for Statistical Computing (2015).

本节要点

- 相关系数测量了两个变量间相互关联的程度。
- 如果变量 v1 的高值随变量 v2 的高值的变化而变化，那么 v1 和 v2 是正相关的。
- 如果变量 v1 的高值与变量 v2 的低值的变化相关联，那么 v1 和 v2 是负相关的。
- 相关系数是一种标准化的度量，因此其值的范围处于 −1（完全负相关）和 +1（完全正相关）之间。
- 如果相关系数为 0，那么表示两个变量间没有相关性。但是注意，数据的随机排列将会随机生成正的或负的相关系数。

1.7.2 拓展阅读

David Freedman、Robert Pisani 和 Roger Purves 合著的 *Statistics*（第 4 版）对相关性进行了很好的介绍。

1.8 探索两个及以上变量

上面介绍的估计量都是我们熟知的，比如均值和方差。计算这些估计量时，我们一次仅查看一个变量，这被称为**单变量分析**。而相关性分析（参见 1.7 节）是比较两个变量间关系的一种重要方法，这是**双变量分析**。本节将介绍一些包含两个及以上变量的估计量及绘图，即**多变量分析**。

主要术语

列联表

一种对两个或两个以上分类变量做计数的表格。

六边形图

一种用于两个数值变量的绘图，图中使用六边形表示记录的组距。

等势线图

一种类似于地形图的绘图，显示了两个数值型变量的密度情况。

小提琴图

一种类似于箱线图的绘图，但是显示的是密度估计量。

与单变量分析一样，双变量分析不仅计算汇总统计量，而且生成可视化的展示。双变量或多变量分析的适用类型取决于数据本身，即数据是数值型数据还是分类数据。

1.8.1 六边形图和等势线（适用于两个数值型变量）

散点图适用于绘制数据量不大的数值型数据，六边形图和等势线也可以绘制数值型数据间的关系。图 1-7 显示了股票日收益的绘图，其中仅包括 750 个数据点。对于具有成千上万乃至上百万条记录的数据集，散点图会过于密集，因此我们需要另一种方式去可视化数据

间的关系。为了解释这个问题，我们以数据集 kc_tax 为例。该数据集包括了对华盛顿州金县（King County）房屋的纳税评估值。为了重点关注数据中的主要部分，我们使用了 subset 函数，去除了数据集中价格非常高并且面积特别小或特别大的房屋。命令如下。

```
kc_tax0 <- subset(kc_tax, TaxAssessedValue < 750000 & SqFtTotLiving>100 &
                  SqFtTotLiving<3500)
nrow(kc_tax0)
[1] 432693
```

图 1-8 是**六边形图**，它显示了金县的房屋面积（平方英尺）与纳税评估值间的关系。六边形图绘制的并非数据点，这样会导致图中显示成一片黑云，我们将记录分组为六边形的组距，并用不同的颜色绘制各个六边形，以显示每组中的记录数。在图中可清晰地看到，房屋面积（平方英尺）和纳税评估值间是正相关的。图中值得关注的一个特征是，在主要云上，隐含有另一片云。这片云所表示的房屋虽然与主要云所表示的房屋具有相同的面积，但是纳税评估值更高。

图 1-8 的绘图是使用 R 语言的 ggplot2 软件包生成的。该软件包是由 Hadley Wickham 研发的 [11]。目前有多个新软件库提供了数据的深层探索性可视化分析功能，ggplot2 就是其中之一，参见 1.8.4 节。

```
ggplot(kc_tax0, (aes(x=SqFtTotLiving, y=TaxAssessedValue))) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient(low="white", high="black") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
```

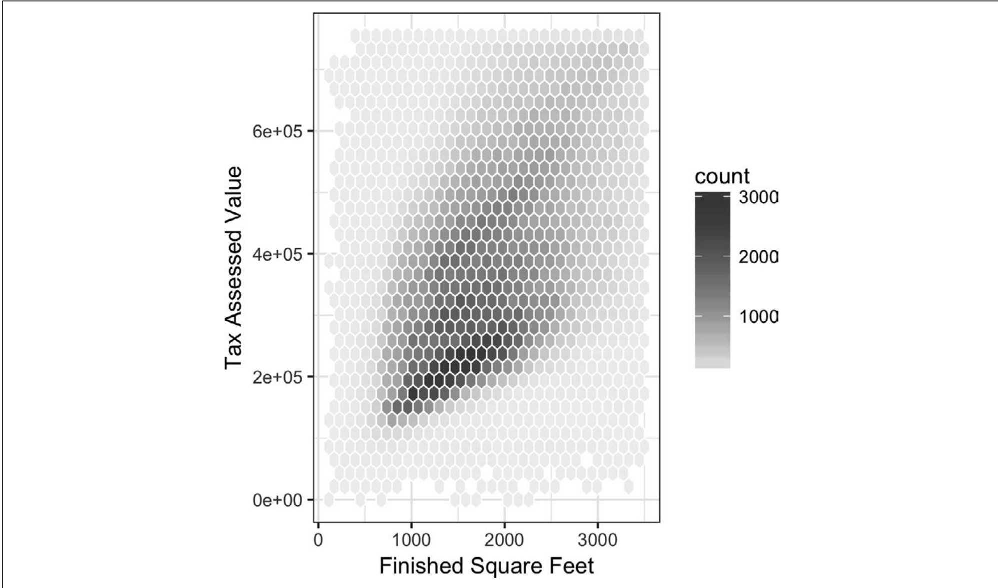


图 1-8：纳税评估值与房屋面积的六边形图

注 11：Wickham, Hadley. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York (2009). ISBN: 978-0-387-98140-6.

图 1-9 是在散点图上绘制了一个等势线图，可视化了两个数值型变量之间的关系。等势线在本质上就是两个变量的地形图。每条等势线表示特定的密度值，并随着接近“顶峰”而增大。图 1-9 显示了类似于图 1-8 中的信息，即在主峰的“北侧”存在第二个峰。图 1-9 也是使用 ggplot2 创建的，其中使用了自带的 geom_density2d 函数。

```
ggplot(kc_tax0, aes(SqFtTotLiving, TaxAssessedValue)) +
  theme_bw() +
  geom_point( alpha=0.1) +
  geom_density2d(colour="white") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
```

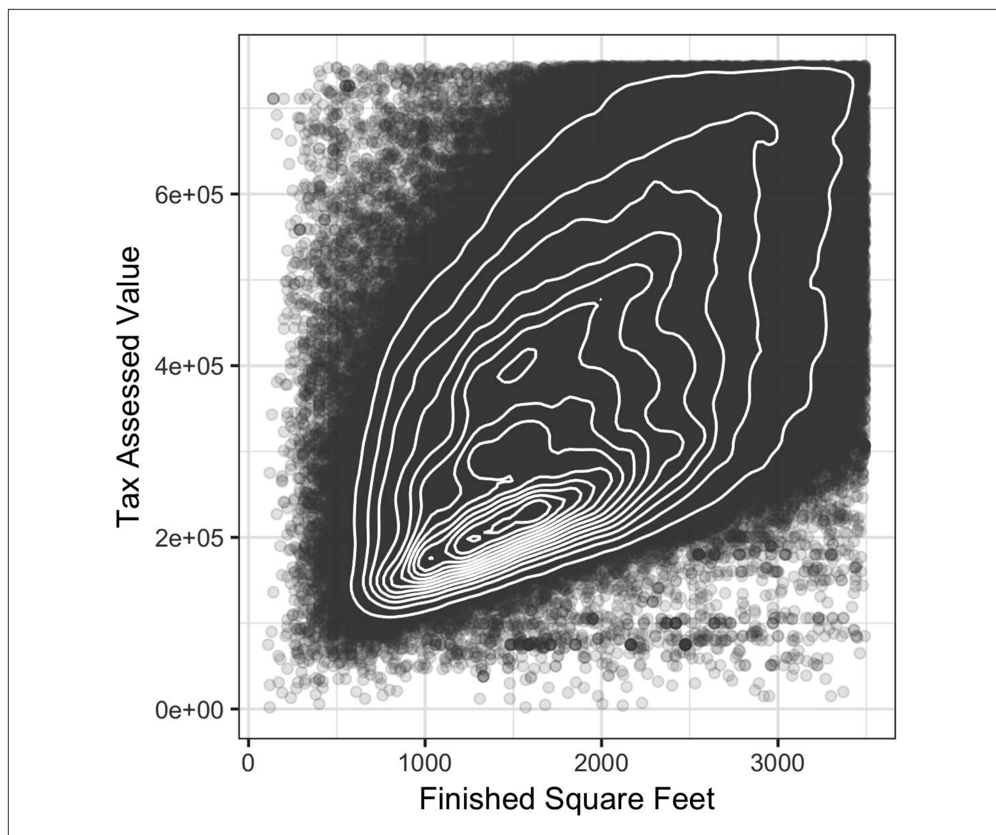


图 1-9：纳税评估值与房屋面积间关系的等势线图

还有其他一些类型的图表，比如**热力图**等，也可显示两个数值型变量间的关系。热力图、六边形图和等势线图所给出的都是二维密度的可视化表示。它们本质上对应的是直方图和密度图。

1.8.2　两个分类变量

对于总结两个分类变量，列联表十分有用，它是一种按分类进行计数的表。表 1-8 显示了

一个列联表，它表示了个人信贷等级与贷款情况之间的关系。数据来自美国 P2P 借贷业务的引领者 Lending Club 公司。信贷等级从 A（高等级）到 G（低等级）不等，贷款情况包括**付清**、**流动**、**延期**和**销账**（预期无法收回剩余贷款）。该表显示了计数情况和按行的百分比情况。与低信贷等级贷款相比，高信贷等级贷款的延期或销账的百分比非常低。列联表中可以只统计计数的情况，也可以包括列百分比和总百分比。最常用的列联表创建工具可能是 Excel 中的数据透视表。在 R 语言中，可以使用 `descr` 软件包中的 `CrossTable` 函数生成列联表。例如，表 1-8 是使用下面的代码生成的。

```
library(descr)
x_tab <- CrossTable(lc_loans$grade, lc_loans$status,
                    prop.c=FALSE, prop.chisq=FALSE, prop.t=FALSE)
```

表1-8：信贷等级和贷款状态的列联表

信贷等级	付清	流动	延期	销账	合计
A	20 715	52 058	494	1588	74 855
	0.277	0.695	0.007	0.021	0.161
B	31 782	97 601	2149	5384	136 916
	0.232	0.713	0.016	0.039	0.294
C	23 773	92 444	2895	6163	125 275
	0.190	0.738	0.023	0.049	0.269
D	14 036	55 287	2421	5131	76 875
	0.183	0.719	0.031	0.067	0.165
E	6089	25 344	1421	2898	35 752
	0.170	0.709	0.040	0.081	0.077
F	2376	8675	621	1556	13 228
	0.180	0.656	0.047	0.118	0.028
G	655	2042	206	419	3322
	0.197	0.615	0.062	0.126	0.007
合计	99 426	333 451	10 207	23 139	466 223

1.8.3 分类数据和数值型数据

一些数值型数据是根据分类变量进行分组的。要可视化地比较此类数据的分布情况，一种简单的方式是使用箱线图（参见 1.5.1 节）。例如，我们可能想要查看各航空公司航班延误的百分比。图 1-10 显示了在一个月内由于航空公司原因所导致航班延误的百分比。

```
boxplot(pct_carrier_delay ~ airline, data=airline_stats, ylim=c(0, 50))
```

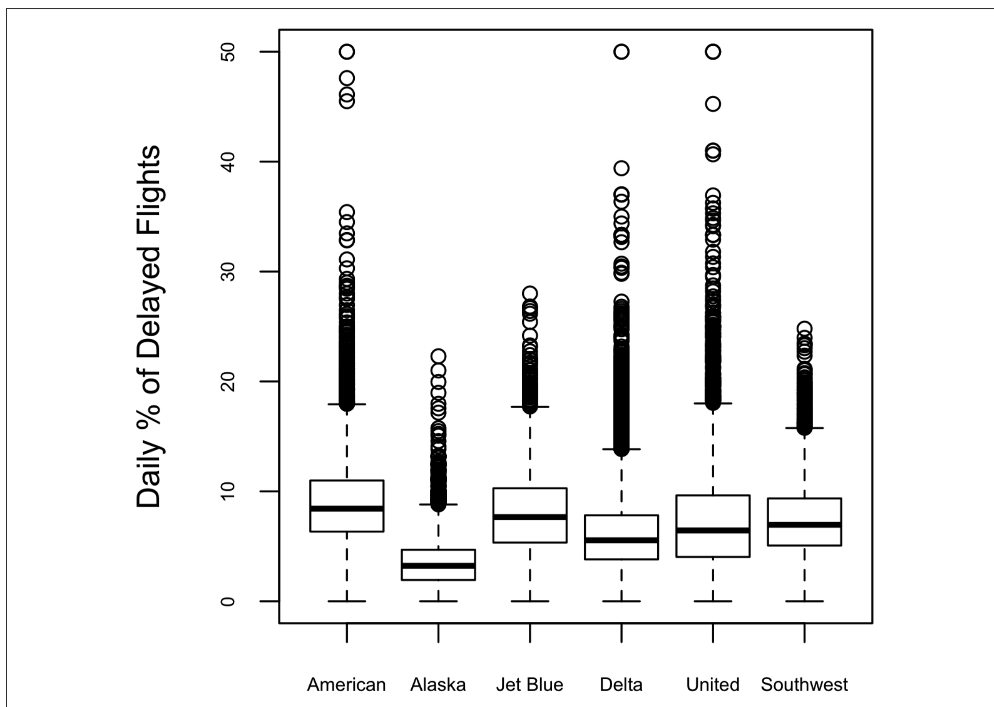


图 1-10：由于航空公司原因所导致航班延误百分比的箱线图

从图中可以看到，阿拉斯加航空公司（Alaska）脱颖而出，延迟最少，而美国航空公司（American）的延迟最多。美国航空公司的下四分位数要高于阿拉斯加航空公司的上四分位数。

小提琴图是箱线图的一种增强表示，最早是由 Hintze 和 Nelson 提出的 [12]。它以 y 轴为密度，绘制密度估计量的情况。绘图中对密度做镜像并反转（即核密度函数），并填充所生成的形状，由此生成了一个类似小提琴的图形。小提琴图的优点是可以显示分布的细微之处，而这种细微之处在箱线图中是难以察觉的。另一方面，箱线图能更清楚地显示数据中的离群值。可使用 `ggplot2` 软件包中的 `geom_violin` 函数创建小提琴图，命令如下。

```
ggplot(data=airline_stats, aes(airline, pct_carrier_delay)) +
  ylim(0, 50) +
  geom_violin() +
  labs(x="", y="Daily % of Delayed Flights")
```

生成的图如图 1-11 所示。小提琴图显示阿拉斯加航空公司的分布聚集于 0 附近，美国达美航空公司（Delta）稍逊之。如果使用箱线图，这一现象并不明显。如果在绘图命令中添加 `geom_boxplot` 函数，那么就可以组合显示小提琴图和箱线图。如果使用了适当的颜色，那么显示效果会更好。

注 12：Hintze, J. and Nelson, “R. Violin Plots: A Box Plot-Density Trace Synergism.” *The American Statistician* 52.2 (May 1998): 181–184.

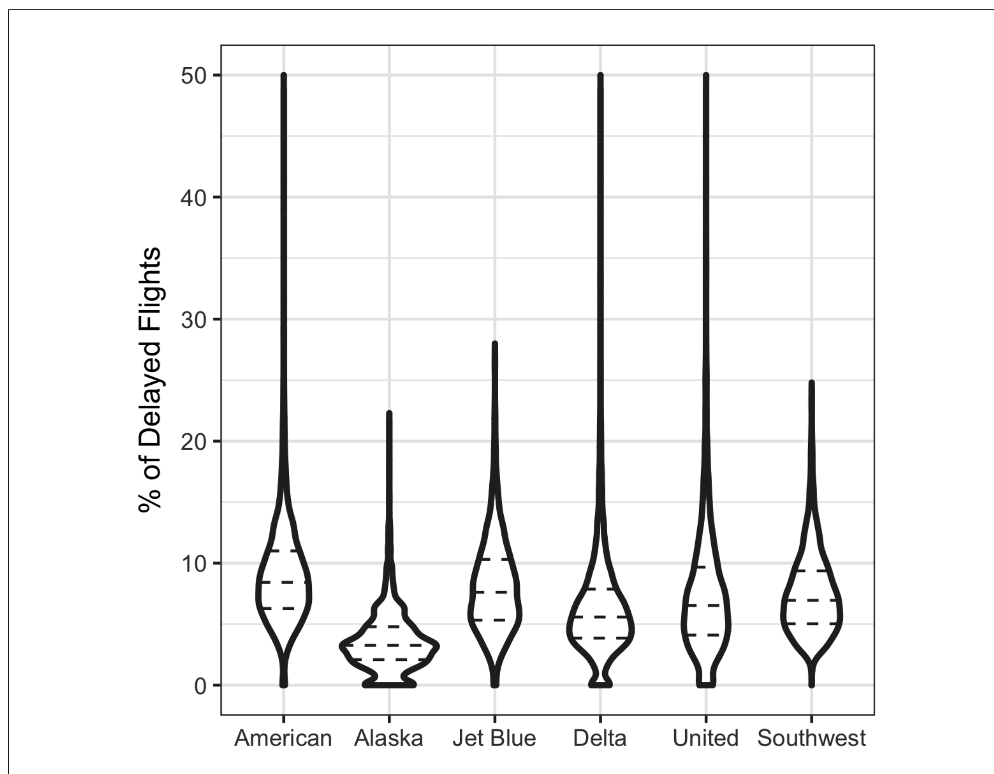


图 1-11：由于航空公司原因所导致航空延误的百分比，图中组合显示了箱线图和小提琴图

1.8.4 多个变量的可视化

比较两个变量所用的图表类型，例如散点图、六边形图和箱线图，完全可以通过**条件**（conditioning）这一概念扩展到多个变量。例如，前面的图 1-8 中显示了房屋面积（单位：平方英尺）和纳税评估值之间的关系。我们发现，一些房屋的每平方英尺纳税评估值看上去更高。进一步研究后，图 1-12 根据邮政编码分别绘制了数据，以比较地段对纳税评估值的影响。从这样的图形中，我们可以更清晰地看到，一些地区（如邮编 98112 和 98105）的纳税评估值比其他地区（如邮编 98108 和 98057）更高。正是这种差异性导致了在图 1-8 中观察到的聚类情况。

我们使用 `ggplot2` 以及**分组**（facet）的概念创建了图 1-12。分组也被称为**条件变量**，在本例中就是邮政编码。

```
ggplot(subset(kc_tax0, ZipCode %in% c(98188, 98105, 98108, 98126)),
         aes(x=SqFtTotLiving, y=TaxAssessedValue)) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient( low="white", high="blue") +
  labs(x="Finished Square Feet", y="Tax Assessed Value") +
  facet_wrap("ZipCode")
```

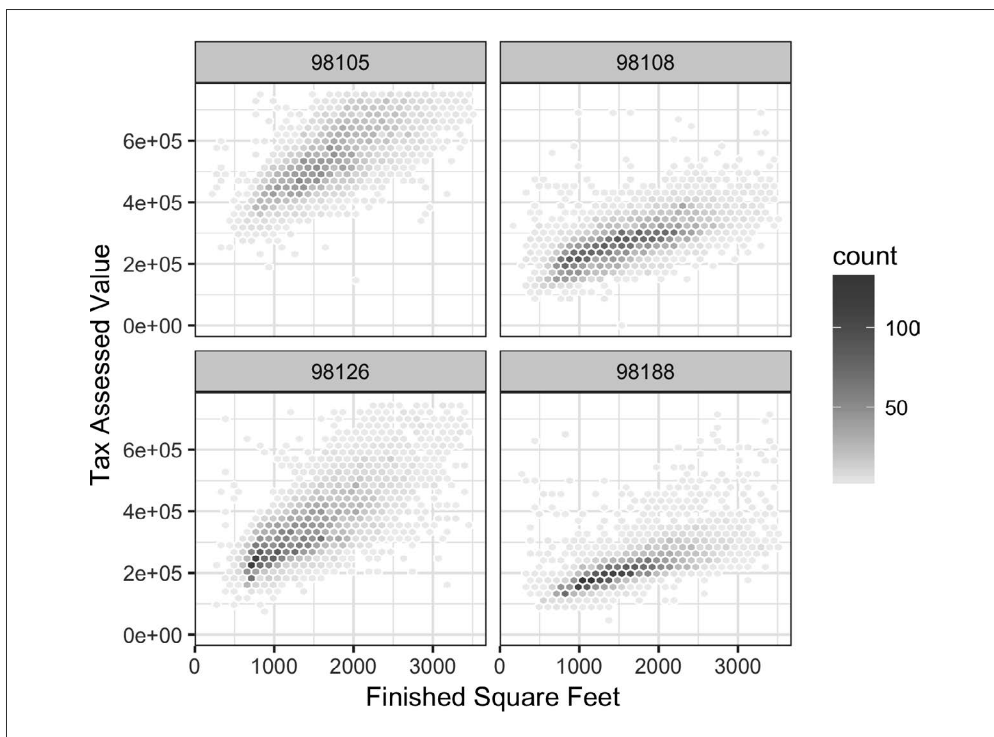


图 1-12：根据邮政编码分组绘制纳税评估值与房屋面积的关系

“条件变量”这一概念最早出现在图形系统的**格子图**中。它是由贝尔实验室的里克·贝克尔、比尔·克利夫兰等人提出的[13]。现在这一理念已扩展到各种现代图形系统中，包括 R 语言的 `lattice`[14] 和 `ggplot2` 软件包，以及 Python 的 `Seaborn`[15] 和 `Bokeh`[16] 模块。条件变量也是 Tableau 和 Spotfire 等商业智能平台的组成部分。随着计算能力的提升，现代可视化平台早已超越了探索性数据分析最初的低水平。但是多年前提出的关键理念和工具依然是这些系统的基础。

本节要点

- 六边形图和等势线图是有用的工具，它们支持以图形方式同时查看两个数值型变量，不会受数据规模的影响。
- 列联表是一种查看两个分类变量计数情况的标准工具。
- 箱线图和小提琴图允许根据分类变量绘制数值型变量。

注 13：Becker, R., Cleveland, W, Shyu, M. and Kaluzny, S.“A Tour of Trellis Graphics”(1996).

注 14：Sarkar, Deepayan. *Lattice: Multivariate Data Visualization with R*. Springer (2008). ISBN 978-0-387-75968-5.

注 15：“Seaborn: statistical data visualization”(2015).

注 16：Bokeh Development Team.“Bokeh: Python library for interactive visualization”(2014).

1.8.5　拓展阅读

- Benjamin Baumer、Daniel Kaplan 和 Nicholas Horton 合著的 *Modern Data Science with R* 一书对“图形的语法”（grammar for graphics，即 `ggplot` 软件包名中的“gg”）进行了很好的介绍。
- Hadley Wickham 撰写的《ggplot2：数据分析与图形艺术》一书提供了一些很好的资源。Hadley Wickham 是 `ggplot2` 的创建者。
- Josef Fruehwald 提供了 `ggplot2` 指南的 Web 资源。

1.9　小结

约翰·图基开创了探索性数据分析。随着探索性数据分析的发展，由统计学所确立的基础业已成为数据科学领域的先导。对于任意基于数据的项目，最重要的第一步都是**查看数据**，这正是探索性数据分析的关键理念所在。通过总结并可视化数据，我们可以对项目获得有价值的洞悉和理解。

本章回顾了一些基本概念，从位置估计和变异性估计等简单度量，到图 1-12 这样的探索多个变量间关系的复杂展示。借助开源社区提供的多种技术和工具集，并结合 R 和 Python 等语言的表达能力，我们得以建立丰富多样的数据探索和分析方式。探索性分析应成为所有数据科学项目的基石。

第 2 章

数据和抽样分布

不少人误以为大数据时代意味着抽样时代的终结。事实上，抽样能够有效地操作一组数据，并且可以最小化偏差。在大数据时代，涌现出了大量质量不一、相关性各异的数据，这增强了人们对于抽样的需求。甚至在大数据项目中，通常也会使用抽样生成并导出预测模型。抽样还被广泛用于定价、Web 处理等各种检验。

本章的理念可以用图 2-1 的模式表述。图中左侧表示总体，统计学假设总体遵循一个潜在的**未知**分布。图的右侧表示**抽样**数据及其经验分布，这是我们唯一可用的。要想根据左侧的图获得右侧的图，我们需要做**抽样**，图中用箭头表示。传统统计学关注的主要是图的左侧部分，即如何对总体运用一些基于强假设的理论。现代统计学已将关注点转移到图的右侧部分，因而也不再需要做出假设。

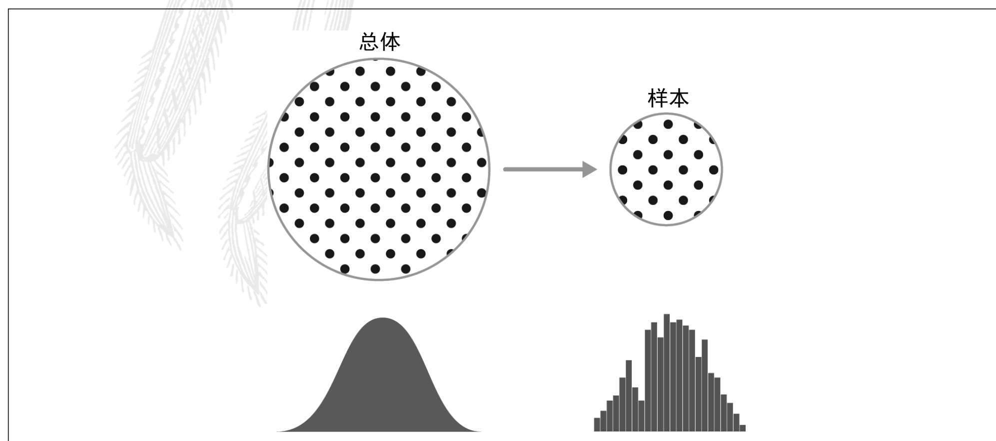


图 2-1：总体与样本

一般而言，数据科学家并不需要操心图中左侧（即总体）的理论本质，而是应聚焦于抽样过程和手中的数据。但有一些特定的情况仍需要他们关注。有些数据是由可建模的物理过程生成的。最简单的例子就是遵循二项分布的抛硬币过程。现实生活中的所有二项分布，例如是否购买、是否存在欺诈、是否点击等，都可以有效地建模为一次抛硬币的过程。当然，一般还需要对硬币正面向上的概率做一定的修正。在此类情况下，我们可以通过对总体的理解，从中获得一些额外的洞见。

2.1 随机抽样和样本偏差

样本是大型数据集的一个子集，统计学家通常将大型数据集称为**总体**。统计学中的总体不同于生物学中所指的总体。在统计学中，总体指的是大量确实存在的数据，但有时也可以是一个理论上的或者构想得到的数据集。

主要术语

样本

大型数据集的一个子集。

总体

一个大型数据集，或是一个构想的数据集。

***N*（或 *n*）**

一般用 N 表示总体的规模，n 表示样本的规模。

随机抽样

从总体中随机抽取元素到样本中。

分层抽样

对总体分层，并在每层中做随机抽样。

简单随机抽样

在不对总体分层的情况下，做随机抽样所得到的样本。

样本偏差

样本对总体做出了错误的解释。

在**随机抽样**过程中，以均等的机会从总体的所有可用成员中抽取，得到一个样本。随机抽样生成的样本被称为**简单随机样本**。抽样可以是**有放回**的，即可以在每次抽取后将所抽取的观测值放回到总体中，并可被随后的抽取重新选中。抽样也可以是**无放回**的，即一个观测值一旦被抽取，就不会参与随后的抽取。

一般情况下，我们在做估计或是根据样本拟合模型时，数据质量的影响要大于数据规模的影响。在数据科学中，数据质量涉及数据的完整性、格式的一致性、整洁性以及单个数据点的准确性。在统计学中，数据质量还涉及抽样的**代表性**这一概念。

一个经典的例子是 1936 年美国《文学文摘》杂志发起的一次民意调查，该调查的结果预

测艾尔弗·兰登将在美国总统选举中战胜富兰克林·罗斯福。《文学文摘》在当时是一份市场份额领先的杂志，此次问卷调查的对象是该期刊的所有订阅者，还额外考虑了一些人，规模合计超过 1000 万人，并预测兰登将取得压倒性胜利。一周后，盖洛普民意调查的创始人乔治·盖洛普也发起了一次民意调查，调查对象只有约 2000 人，但准确地预测了罗斯福会取得胜利。两次调查的差异在于调查对象的选择。

《文学文摘》侧重于调查对象的数量，忽视了选择方法。他们的调查对象是那些社会经济地位相对较高的人群（即该杂志的订阅者，以及那些在当时有电话和汽车等奢侈品的人，他们是市场营销人员的目标）。这导致了调查结果中存在**样本偏差**，即样本以某种有意义的非随机方式，不同于其想要代表的大规模总体。**非随机性**（nonrandom）这一术语非常重要，因为几乎任何样本都无法准确地表示总体，即便是随机抽样也做不到。一旦差异具有意义，就会发生样本偏差。如果其他样本也使用了同样的抽取方式，那么也会存在样本偏差。



自选择抽样偏差（self-selection sampling bias）

在 Yelp 等社交媒体上，我们能看到一些对餐馆、酒店、咖啡馆等的评论。这些评论容易产生偏差，因为提交评论的人并非随机选取的。他们写评论是基于一定的出发点的，这将导致**自选择偏差**的产生。有意向撰写评论的人，很可能是那些获得了不好体验的人，也可能是一些与商家有关联的人，或者就是与没有发表评论者不同的一类人。注意，在将一个商家与类似的商家做对比时，尽管自选择样本或许并未可靠地表明事情的真实状态，但它们依然是更为可靠的，因为对比的双方都存在同样的自选择偏差。

2.1.1 偏差

统计偏差是一些系统性的测量误差或抽样误差，它是在测量或抽样过程中产生的。我们应严格区分由随机选取所导致的误差和由偏差所导致的误差。以开枪射击一个目标这一物理过程为例。并非每次射击都能击中绝对意义上的靶心，或者说很少能击中。虽然无偏过程也会产生误差，但所产生的误差是随机的，并且不会强烈地趋向于任意方向，如图 2-2 所示。图 2-3 给出的是一个有偏过程的结果，在 x 轴和 y 轴方向上，不仅存在着随机误差，还存在着偏差。射击点趋向于落在右上象限部分。

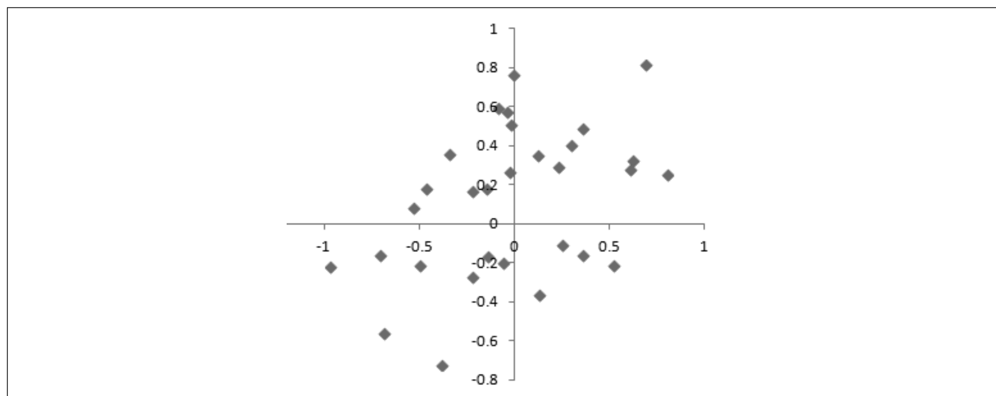


图 2-2：一支瞄准正常的枪射击情况的散点图

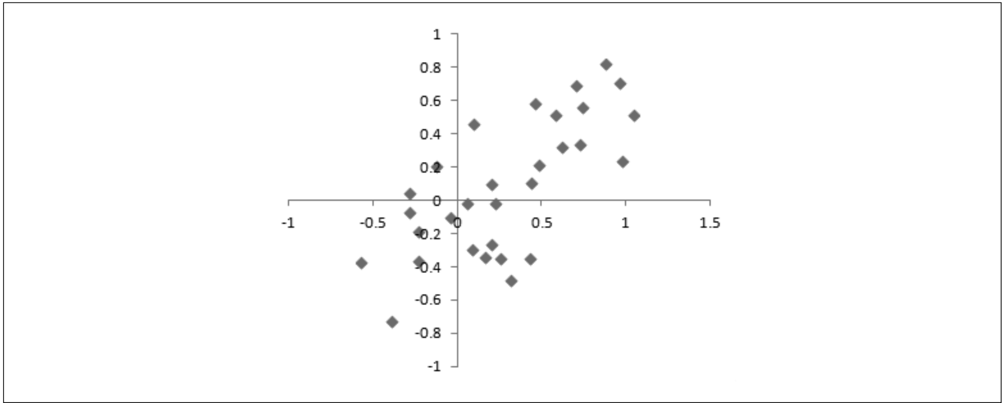


图 2-3：一支瞄准有偏差的枪射击情况的散点图

偏差有多种形式，它也许是可以观察到的，也可能是不可见的。如果结果确实表示存在偏差（例如，通过参考基准或实际值），这通常表明我们指定了不正确的统计学或机器学习模型，或是漏掉了某个重要的变量。

2.1.2 随机选择

为了避免出现导致《文学文摘》预测兰登在选举中战胜罗斯福这样的样本偏差问题，乔治·盖洛普（图 2-4）采用了一种更科学的方法来得到可以代表美国选民的样本。如今，实现样本代表性的方法有很多，所有这些方法的核心都是**随机抽样**。



图 2-4：乔治·盖洛普，因《文学文摘》的“大数据”失败而名声大噪

随机抽样并不容易实现，关键在于如何正确地定义可访问的总体。假设我们想要生成客户的一个代表性形象，并且需要执行一次试点客户调查。调查要具有代表性，但是所需的工作强度极大。

首先，我们需要定义客户是谁。我们可以选择购买金额大于零的所有客户记录。那么是否要考虑过去所有的客户？是否要考虑退款情况？是否要考虑内部测试购买情况？是否需要考虑经销商、结算代理人和客户？

下一步，我们要指定抽样过程。抽样可以是“随机选取 100 名客户”。当涉及对某个数据流的抽样时，如实时客户交易、Web 访问者等，时间上的考虑很重要，例如工作日上午十点的 Web 访问者可能不同于周末晚上十点的 Web 访问者。

采用**分层抽样**时，我们将总体分成多个**层**，并在每一层中做随机抽样。例如，在一次政治民意调查中，可能需要了解美国白人、非裔美国人和拉美裔美国人的选举倾向。如果我们对总体做一次基本的随机抽样，得到的样本中可能非裔和拉美裔美国人人数过少。因此在分层抽样中，需要对不同的层赋予不同的权重，以生成对等的抽样规模。

2.1.3 数据规模与数据质量：何时规模更重要

在大数据时代，令人惊讶的是，有时数据规模越小，结果反而更好。在随机抽样上花费些时间和精力，不仅可以减小偏差，还能让我们更关注于数据探索和数据质量。例如，在缺失的数据和离群值中，可能包含了一些有用的信息。要从上百万条记录中查找缺失值或评估离群值，成本可能会非常高，但是对于具有数千条记录的样本，这些事情则是完全可行的。此外，如果数据量过大，也无法开展数据绘图和人工检测。

那么，在什么情况下需要大量的数据呢？

Google 检索查询请求，就是一个体现大数据价值的经典场景，其中数据不仅规模很大，而且十分稀疏。如果以每个词为列、每个搜索查询为行，这样可以构建一个矩阵。矩阵中每个单元的值为 0 或 1，表示相应的查询中是否包含对应的词汇。我们的目标是对一个查询给出一个最优的搜索目标。但是，英语中有 15 万多个单词，而 Google 每年会处理大约一万亿次查询。这生成了一个规模非常巨大的矩阵，矩阵中大量单元的值为 0。

这是一个真正的大数据问题。只有积累了如此巨大规模的数据后，Google 才能为大部分查询提供有效的搜索结果。积累的数据越多，查询结果越好。对于一些常见的搜索词，并不存在问题，因为对于在某一时刻非常流行的主题，我们可以很快发现有效数据。而如何对多种多样的检索查询返回详细且有用的结果，甚至包括那些出现频数只有百万分之一的查询，这正是现代搜索技术的真正价值所在。

例如，我们要查询短语“里奇 · 里卡多和小红帽”。在互联网出现的早期，查询的返回结果可能是乐队领袖里奇 · 里卡多、他主演的电视剧《我爱露西》，以及儿童剧《小红帽》。但是现代搜索引擎已具有数万亿条查询检索记录，因此检索查询可以精确地返回《我爱露西》中的一集，里卡多在其中用英语和西班牙语为他襁褓中的儿子讲述《小红帽》的故事。

注意，确实**相关的**记录可能需要达到数千条才是有效的。这里所说的“相关”，指的是记录中出现了查询词或类似内容（连同有关人们最终点击的链接的信息）。但是，为了获得这样的相关记录，可能需要处理数万亿条数据。当然，随机抽样并不会有作用。参见 2.8 节。

2.1.4 样本均值与总体均值

总体中的样本均值一般用符号 $\bar{x}$ 表示，而总体的均值一般用 μ 表示。为什么要区分这两者？这是因为样本的信息是可以观测到的，而大规模的总体的信息通常获取自规模较小的

样本。统计学家喜欢从符号上对两者加以区分。

> **本节要点**
>
> - 即便是在大数据时代，随机抽样依然是数据科学家的一种重要手段。
> - 由于测量或观测不能代表总体而出现系统性误差时，就会产生偏差。
> - 数据的质量通常比数量更重要，而随机抽样可以降低偏差，提高数据的质量（否则，实现成本可能很高）。

2.1.5　拓展阅读

- 在 *The Sage Handbook of Online Research Methods* 一书中，Ronald Fricker 撰写了一章“Sampling Methods for Web and E-mail Surveys”，其中对抽样过程的介绍十分有用。该章综述了对随机抽样方法的一些改进，基于成本或可行性的实际考虑，这些改进经常被使用。
- 在 Capital Century 网站上可以看到有关《文学文摘》调查失败的介绍。

2.2　选择偏差

尤吉·贝拉（Yogi Berra）有一句名言：“如果你不知道自己在寻找什么，那么努力去寻找吧，终会发现它。”

选择偏差是指以一种可导致误导性或短暂性结论的方式，有选择性地选取数据的操作。选择偏差可能是有意而为之，也可能是无意识的。

> **主要术语**
>
> **偏差**
>
> 　　系统性误差。
>
> **数据窥探**
>
> 　　为得到感兴趣的结果，在数据中做大量的查找。
>
> **大规模搜索效应**
>
> 　　由于重复的数据建模，或使用大量的预测变量对数据建模所导致的偏差或非可重现性。

如果我们指定一个假设，并使用设计良好的实验去验证该假设，就能得到具有高置信度的结论。但实际情况往往并非如此。人们通常只是查看可用的数据，并试图识别数据中的模式。但模式是真实的，还是仅仅是**数据窥探**（即广泛地探查数据，直至发现我们感兴趣的现象）的结果？在统计学家中存在着一个说法：“如果我们拷问数据的时间足够长，那么它迟早会招供。”

通过实验验证一个假设所得到的现象，与通过研判可用数据而发现的现象，这两者之间存在着差别。下面我们通过一个实验给出解释。

假设有人说他能做到抛硬币连续十次正面向上。我们想要挑战他，这就相当于做一次实验。如果他继续抛十次硬币，依然连续正面向上，显然这只能归因于他具有某种特异功能，因为抛硬币连续十次正面向上的概率大约是千分之一。

现在，假设在一个体育场中有两万名观众，我们通过播音员要求全体两万人一起抛十次硬币。如果有人做到了连续十次正面向上，就站出来。我们会看到，整个体育场中很可能**有人**会做到连续十次正面向上。这一事件的概率非常高，甚至会高于 99%，即 1 减去没有人得到十次正面向上的概率。显然，我们事后从所有人中选取能做到十次正面向上的人，并不意味着他们具有任何特异功能，这更像是运气使然。

反复地查看大规模数据集是数据科学中的一个关键价值主张，所以我们需要关注选择偏差问题。数据科学家特别关注的一种选择偏差形式，就是被约翰 · 埃德（John Elder）称为**大规模搜索效应**的问题。约翰 · 埃德是美国 Elder 研究机构的创始人，该机构是一家广受关注的数据挖掘咨询公司。如果在大规模数据集上反复运行不同的模型，并提出不同的问题，我们肯定能发现一些有意思的现象。但是我们所发现的结果是否的确具有意义？还是仅是一些离群值？

为了避免这一问题，我们可以使用验证集（holdout set）去验证结果的性能，有时可能需要多个验证集。埃德倡议使用一种被称为**目标混洗**（target shuffle）的方法。该方法在本质上就是一种置换检验，验证由数据挖掘模型所预测的关联关系的合法性。

在统计学中，除了大规模搜索效应之外，选择偏差的典型形式还包括**非随机抽样**（参见**抽样偏差**）、主观随机挑选（cherry-picking）数据、选取突出特定统计效应的时间间隔，以及在结果看上去“具有意义”时停止实验。

2.2.1　趋均值回归

趋均值回归指对同一变量做连续测量时出现的一种现象，即在极端观测值后，会出现更趋向于中心的观测值。对极值给予特殊的关注和意义，会导致某种形式的选择偏差。

“当年的新秀会在第二年表现低迷。”这是广大体育迷们耳熟能详的一个现象。从某个赛季开始职业生涯的新运动员中，总会有个人的成绩好于其他所有人。但是在第二年，“当年的新秀”的成绩通常会不如上一年。为什么会这样呢？

几乎所有主要的体育运动，至少是打球或冰球，运动员的整体表现取决于两个关键因素。

- 技能
- 运气

趋均值回归是由某种形式的选择偏差所导致的。在选取运动成绩最好的新秀时，技能和好运气可能会同时发挥作用。而在下一个赛季，尽管该运动员的技能依旧，但运气却在很多情况下并非如此。因此他的成绩会下滑，即产生倒退。该现象最早是 1886 年由弗朗西斯 · 加尔顿发现的[1]。在撰写论文时，他将此现象与遗传倾向联系在一起。例如，如果父亲个子

注 1：Galton, Francis.“Regression towards mediocrity in Hereditary stature.”*The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246-273. JSTOR 2841583.

很高，那么子女的身高趋向低于父亲，如图 2-5 所示。

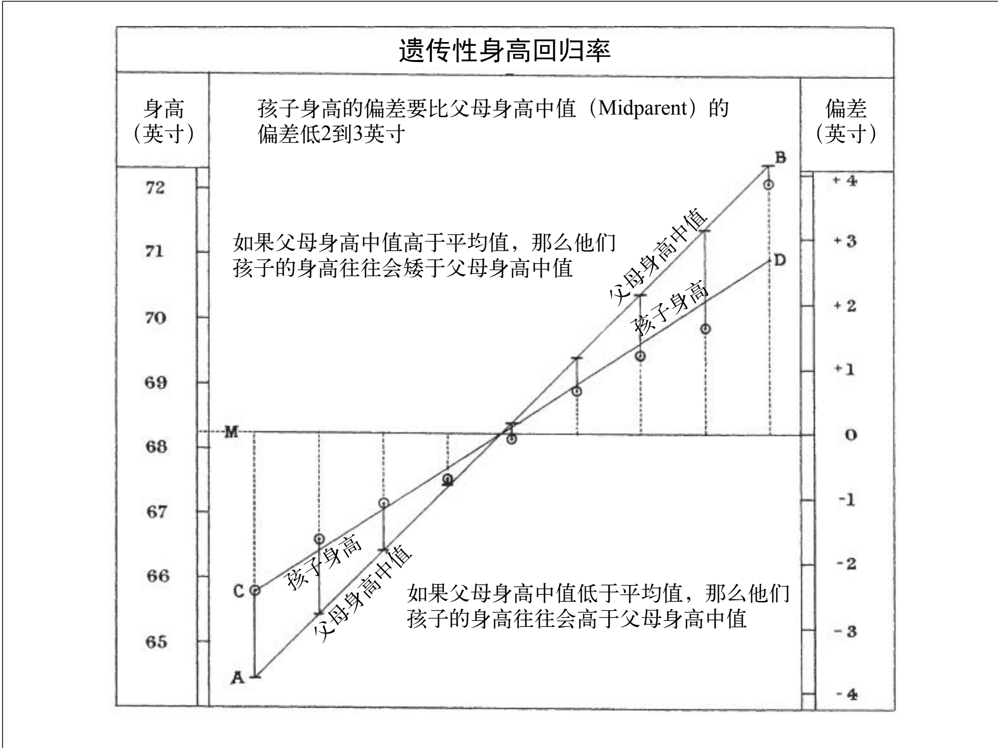


图 2-5：加尔顿的研究提出了趋均值回归现象



从“回退”意义上看，趋均值回归完全不同于统计建模方法的线性回归。线性回归用于估计预测变量和输出变量间的线性关系。

本节要点

- 指定一个假设，然后遵循随机化和随机抽样的原则收集数据，可以确保不会产生偏差。
- 所有其他类型的数据分析都有产生偏差的风险，风险来自数据的采集和分析过程，包括在数据挖掘中反复地运行模型、在研究中窥探数据，以及事后选取有意义的事件。

2.2.2 拓展阅读

- Christopher J. Pannucci 和 Edwin G. Wilkins 在其论文“Identifying and Avoiding Bias in Research”中，对研究中可能会引入的各种偏差（包括选择偏差）进行了综述。该论文发表在 *Plastic and Reconstructive Surgery* 2010 年 8 月刊上。

- Michael Harris 的文章“Fooled by Randomness Through Selection Bias”从一个股票交易人士的角度，对股票市场交易中所考虑的选择偏差问题做了综述。

2.3　统计量的抽样分布

统计量的**抽样分布**指从同一总体中抽取多个样本时，一些样本统计量的分布情况。经典统计学主要关注如何从小样本推导更大总体的情况。

> **主要术语**
>
> **样本统计量**
>
> 对抽取自大规模总体中的样本做计算，所得到的一些度量值。
>
> **数据分布**
>
> 单个**值**在数据集中的频数分布。
>
> **抽样分布**
>
> 一个**样本统计量**在多个样本或重抽样中的频数分布。
>
> **中心极限定理**
>
> 当样本的规模增大时，抽样分布呈正态分布的趋势。
>
> **标准误差**
>
> 多个样本间样本**统计量**的变异性（标准偏差）。不要与**标准偏差**混淆，后者指的是个体数据**值**间的变异性。

我们从总体中抽取样本，通常是为了测量某个样本统计量，或是使用统计学或机器学习模型进行建模。鉴于估计量或模型是基于某个样本的，因此其中可能存在误差，也可能会由于抽取样本的不同而有所差异。我们需要了解这种差异究竟如何，即我们的主要关注点在于**抽样的变异性**。如果有大量的数据，那么我们可以从中抽取更多的样本，进而直接观察样本统计量的分布情况。只要数据易于获取，那么我们一般会使用尽可能多的数据去计算估计量或拟合模型，而非总是使用从总体中抽取更多样本的方法。



区分单个数据点的分布（即**数据分布**）和样本统计量的分布（即**抽样分布**）非常重要。

通常，样本统计量（如均值等）的分布要比数据本身的分布更加规则，分布的形状更趋向于正态分布的钟形曲线。统计所基于的样本规模越大，上面的观点就愈发成立。此外，样本的规模越大，样本统计量的分布就越窄。

下面我们用一个例子来解释这一观点。本例中使用的数据来自向 Lending Club 公司申请贷款者的年收入数据（对于数据的详细描述，参见 6.1.1 节）。我们对数据做三次抽样，得到的三个样本分别为：具有 1000 个值的样本、取 5 个数据均值的 1000 个均值样本，以及取

20 个数据均值的 1000 个均值样本。然后我们绘制每个样本的直方图，如图 2-6 所示。

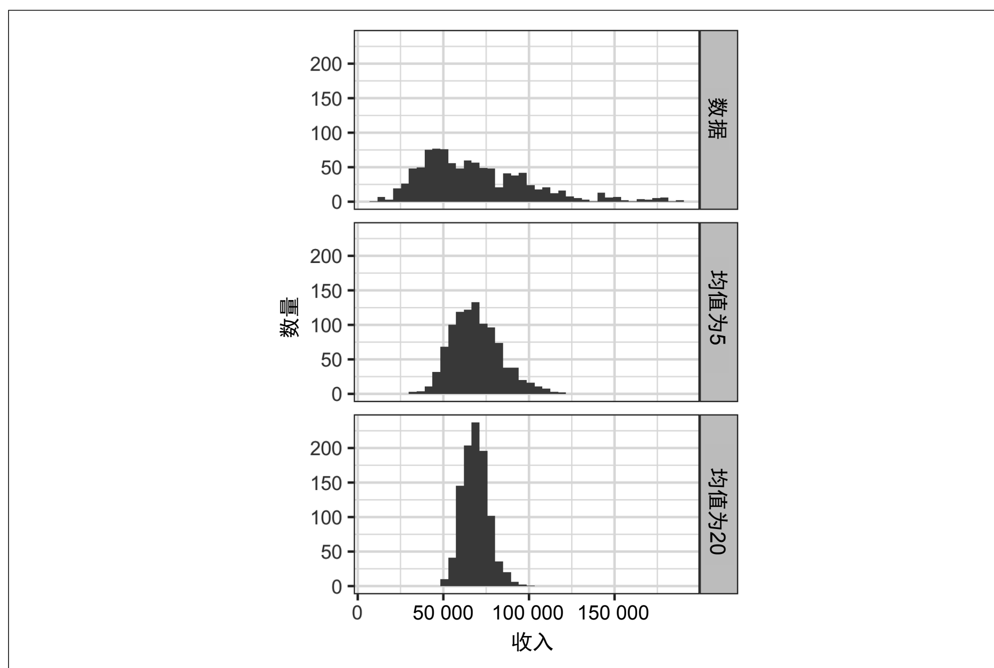


图 2-6：贷款申请者年收入样本的直方图。从上到下的样本依次为：1000 名贷款申请者样本（上，即 $n = 1$）、$n = 5$ 的 1000 个均值样本（中），以及 $n = 20$ 的 1000 个均值样本（下）

单个数据值样本的直方图的分布很宽泛，并且向更高值处偏斜，这与对收入数据的预期一致。$n = 5$ 和 $n = 20$ 的均值样本的直方图表现出一种愈加紧凑的趋势，并且形状更趋向于钟形。下面给出生成上面直方图的 R 代码，其中使用了可视化软件包 `ggplot2`。

```
library(ggplot2)
# 做一次简单随机抽样
samp_data <- data.frame(income=sample(loans_income, 1000),
                        type='data_dist')
# 对5个数据的均值做抽样
samp_mean_05 <- data.frame(
  income = tapply(sample(loans_income, 1000*5),
                  rep(1:1000, rep(5, 1000)), FUN = mean),
  type = 'mean_of_5')
# 对20个数据的均值做抽样
samp_mean_20 <- data.frame(
  income = tapply(sample(loans_income, 1000*20),
                  rep(1:1000, rep(20, 1000)), FUN = mean),
  type = 'mean_of_20')
# 将抽样结果绑定到一个data.frames对象，并转化为因子类型
income <- rbind(samp_data, samp_mean_05, samp_mean_20)
income$type = factor(income$type,
                     levels=c('data_dist', 'mean_of_5', 'mean_of_20'),
                     labels=c('Data', 'Mean of 5', 'Mean of 20'))
```

```
# 绘制直方图
ggplot(income, aes(x=income)) +
  geom_histogram(bins=40) +
  facet_grid(type ~ .)
```

2.3.1　中心极限定理

上例中的现象被称为**中心极限定理**。该定理指出，即便原始总体不符合正态分布，但是只要样本的规模足够大，并且数据并非在很大程度上偏离正常值，那么从多个样本得到的均值将会呈现出我们所熟知的钟形正态曲线（参见 2.6 节）。在使用抽样分布做推理时，即置信区间和假设检验中，中心极限定理允许我们使用 t 分布这样的近似正态公式。

中心极限定理在传统的统计学教科书中得到了大量的关注，因为它是支持假设检验和置信区间的底层机制，而这些内容本身就占据了教科书的一半篇幅。数据科学家应该了解这一点，但是鉴于在数据科学中，任意场景都能使用**自助法**（bootstrap）解决问题，很少正式地使用假设检验和置信区间，因此中心极限定理并非处于数据科学实践的中心位置。

2.3.2　标准误差

标准误差是一种单变量度量，它总结了单个统计量抽样分布的变异性。标准误差可以根据样本值的标准偏差 s 和样本规模 n，使用基于统计学的方法进行估计，公式如下。

$$\text{标准误差} = \frac{s}{\sqrt{n}}$$

正如我们在图 2-6 中所观察到的，标准误差会随样本规模的增大而减小。有时，我们称标准误差与样本规模间的关系为 ***n* 的平方根**规则。如果要使标准误差减小一半，那么样本规模应该增大四倍。

标准误差计算公式的合理性源于中心极限定理（参见 2.3.1 节）。事实上，我们不必依靠中心极限定理来理解标准误差。下面的方法可用于测量标准误差。

(1) 从总体中抽取一些全新的样本；
(2) 对于每个新样本，计算统计量，例如均值；
(3) 对第 2 步计算得到的统计量，计算其标准偏差，以此作为对标准误差的估计。

但是在实践中，通过采集新样本去估计标准误差的方法通常并不可行，从统计意义上看也存在很大的浪费。幸运的是，我们完全不需要抽取全新的样本，而可以使用**自助法**进行重抽样（参见 2.4 节）。在现代统计学中，自助法已成为估计标准误差的标准方法。自助法几乎适用于所有的统计量，它不依赖于中心极限定理或其他的分布假设。



标准偏差与标准误差

不要将标准偏差和标准误差混为一谈。标准偏差测量的是单个数据点的变异性，而标准误差测量的是抽样度量的变异性。

本节要点

- 样本统计量的频数分布表明了度量在各个不同抽样间的变化情况。
- 抽样分布可以使用自助法估计，也可以通过依赖于中心极限定理的公式计算得到。
- 标准误差是一个关键的度量，它汇总了抽样统计量的变异性。

2.3.3 拓展阅读

David Lane 的统计学在线多媒体资源提供了一个有用的模拟环境。你可以选择抽样统计量、样本规模和迭代次数，并且可以将结果频数分布可视化为直方图。

2.4 自助法

要估计统计量或模型参数的抽样分布，一个简单而有效的方法是，从样本本身中有放回地抽取更多的样本，并对每次重抽样重新计算统计量或模型。这一过程被称为**自助法**。自助法无须假设数据或抽样统计量符合正态分布。

主要术语

自助样本（bootstrap sample）

从观测数据集中做有放回的抽取而得到的样本。

重抽样

在观测数据中重复抽取样本的过程，其中包括自助过程和置换（混洗）过程。

从概念上看，我们可以这样理解自助法：将原始样本复制成千上万次，得到一个假想的总体，其中包括了原始样本中的全部信息，只是规模更大。然后我们从这一假想总体中抽取样本，用于估计抽样分布。自助法的理念如图 2-7 所示。

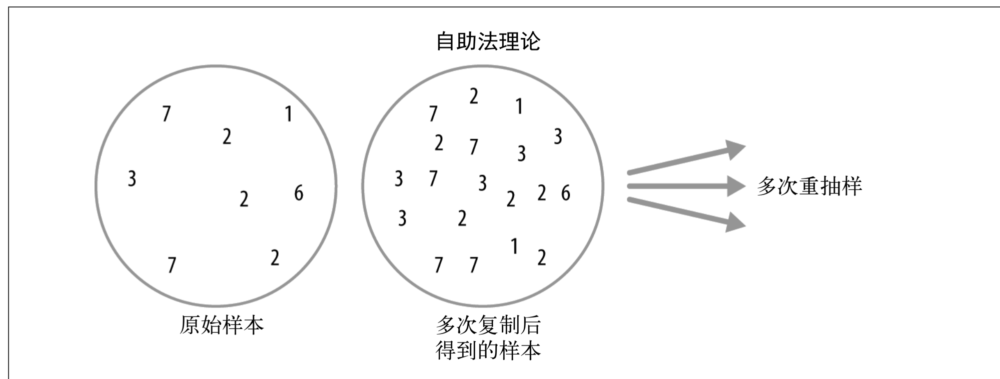


图 2-7：自助法的理念

在实践中，完全不必真正地多次复制样本。只需在每次抽取后，将观测值再放回总体中，即**有放回地抽样**。这一方式有效地创建了一个无限的总体，其中任意一个元素被抽取的概率在各次抽取中保持不变。使用自助法对规模为 n 的样本做均值重抽样的算法实现如下。

(1) 抽取一个样本值，记录后放回总体。
(2) 重复 n 次。
(3) 记录 n 个重抽样的均值。
(4) 重复步骤 1 ～ 3 多次，例如 r 次。
(5) 使用 r 个结果：
  a. 计算它们的标准偏差（估计抽样均值的标准误差）；
  b. 生成直方图或箱线图；
  c. 找出置信区间。

我们称 r 为自助法的迭代次数，r 的值可任意指定。迭代的次数越多，对标准误差或置信区间的估计就越准确。上述过程的结果给出了样本统计量或估计模型参数的一个自助集，可以从该自助集查看统计量或参数的变异性。

R 语言的 `boot` 软件包将上述步骤组合成一个函数。例如，下面的代码实现将自助法用于借款者的收入数据。

```
library(boot)
stat_fun <- function(x, idx) median(x[idx])
boot_obj <- boot(loans_income, R = 1000, statistic = stat_fun)
```

函数 `stat_fun` 计算索引 `idx` 所指定样本的中位数，结果如下。

```
Bootstrap Statistics :
    original   bias    std. error
t1*    62000 -70.5595    209.1515
```

从结果中可看到，中位数的初始估计是 62 000 美元。自助法分布显示，估计量的偏差约为 −70 美元，标准误差约为 209 美元。

自助法也可用于多变量数据。这时该方法使用数据行作为抽样单元，如图 2-8 所示，进而可在自助数据上运行模型，估计模型参数的稳定性（或变异性），或是改进模型的预测能力。我们也可以使用分类和回归树（也称**决策树**）在自助数据上运行多个树模型，并平均多个树给出的预测值（或是使用分类，并选取多数人的投票），这通常要比使用单个树的预测性能更好。这一过程被称为 Bagging 方法。Bagging 一词是 bootstrap aggregating（自助法聚合）的缩写，参见 6.3 节。

自助法反复重抽样的概念十分简单。在经济学家和人口学家朱利安·西蒙（Julian Simon）于 1969 年出版的教科书 *Basic Research Methods in Social Science*[2] 中，汇总了多个重抽样的例子，其中也包括一些自助法的例子。但是，反复重抽样的计算量很大，在计算能力广泛可用之前，它不是一种可行的方法。该技术在 20 世纪 70 年代末 80 年代初才由斯坦福大

注 2：Simon, J. L., & Burstein, P. (1969). *Basic Research Methods in Social Science: The Art of Empirical Investigation*. Random House.

学统计学家布拉德利·埃弗龙（Bradley Efron）命名。当时他在多份学术期刊文章[3]以及一本著作[4]中使用了“自助法”一词。该技术在那些使用统计学方法的非统计学研究人员中得到了尤为广泛的应用，主要用于在数学上不具备解决方法的一些度量或模型。尽管均值的抽样分布方法在 20 世纪 80 年代就已经确立了，但当时对其他度量的抽样分布方法依然尚未确立。自助法还可用于确定抽样的规模，它通过实验查看不同的 n 值对抽样分布的影响。

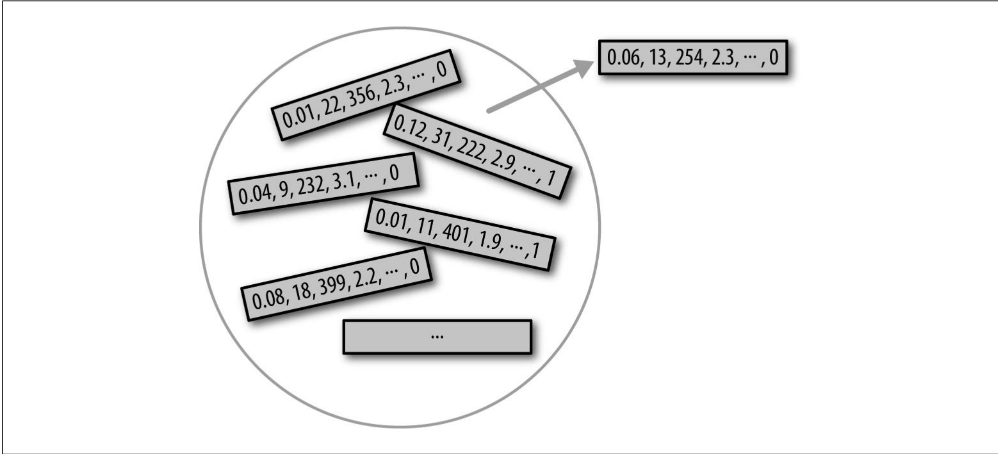


图 2-8：多变量自助法抽样

自助法被首次提出时，曾受到了大量的怀疑，因为它太神奇了。这些怀疑都源于对自助法目标的误解。



自助法并不补偿小规模样本。它不创建新的数据，也不会填补已有数据集中的缺口。它只会告知我们，在从原始样本这样的总体中做抽取时，大量额外的样本所具有的行为。

2.4.1　重抽样与自助法

正如上节所介绍的，有时**重抽样**这个词等同于**自助法**。在更多情况下，**重抽样**还包括置换过程（参见 3.3.1 节）。置换过程组合了多个样本，并且抽样可能是无放回的。但是在任何情况下，**自助法**都是指对观测数据集做有放回的抽样。

注 3：Bradley Efron (1979). Bootstrap Methods: Another Look at the Jackknife. The Annals of Statistics. 7(1):1–26.

注 4：Efron, B. (1982). *The Jackknife, the Bootstrap, and Other Resampling Plans*. Society of Industrial and Applied Mathematics CBMS-NSF Monographs.

本节要点

- 自助法（即对数据集做有放回的抽样）是一种评估样本统计量变异性的强大工具。
- 自助法可以类似的方式应用于各种场景中，无须深入探究抽样分布的数学近似。
- 自助法可以在不使用数学近似的情况下，估计统计量的抽样分布。
- 用于预测模型时，聚合多个自助样本的预测（即 Bagging 方法），要优于使用单个模型的预测。

2.4.2 拓展阅读

- Bradley Efron 和 Robert Tibshirani 合著的 *An Introduction to the Bootstrap* 是首本专门介绍自助法的图书。该书目前依然广为阅读。
- Peter Hall 发表于 *Statistical Science* 2003 年 5 月刊（第 18 卷第 2 期）上的论文“A Short Prehistory of the Bootstrap”，从多个角度对自助法进行了综述，其中介绍了朱利安 · 西蒙于 1969 年首次发表的自助法。
- 在 Gareth James 等人撰写的《统计学习导论：基于 R 应用》一书中，有几节专门介绍自助法，尤其是 Bagging 方法。

2.5 置信区间

要了解一个样本估计量中潜在的误差情况，除了使用前文介绍的频数表、直方图、箱线图和标准误差等方法外，还有一种方法是**置信区间**。

主要术语

置信水平

以百分比表示的置信区间。该区间是从同一总体中以同一方式构建的，可以包含我们感兴趣的统计量。

区间端点

置信区间的两端。

不确定性当然不受人待见。人们（尤其是专家）很少说：“我不知道。”分析人员和管理者虽然会承认不确定性的存在，但是很少会过于信任以单一数值呈现的估计量，即**点估计**。为了解决这一普遍性问题，我们可以使用一个范围而不是单一的值去表示估计量。统计抽样原理是置信区间的实现基础。

置信区间通常以覆盖程度的形式给出，表示为（高）百分比，例如 90% 或 95%。对 90% 置信区间的一种理解方式是，该区间涵盖了样本统计量自助抽样分布中间 90% 的部分（参见 2.4 节）。更通用的理解是，在采用类似抽样过程的情况下，样本统计量的 $x\%$ 置信区间，表明该区间平均在 $x\%$ 的情况下包含类似的样本估计量。

给定样本规模 n，并指定了一个感兴趣的样本统计量，计算自助法置信区间的算法如下。

(1) 从数据中有放回地抽取规模为 n 的随机样本（重抽样）。
(2) 记录重抽样中感兴趣的统计量。
(3) 多次重复步骤 1 ～ 2，例如 r 次。
(4) 对于 x% 置信区间，从分布的两端分别对 r 个重抽样结果切尾 [(1−[x/100])/2]%。
(5) 切尾点就是 x% 自助法置信区间的区间端点。

图 2-9 显示了对于规模为 20、均值为 57 573 美元的样本，申请贷款者的年收入均值的 90% 置信区间。

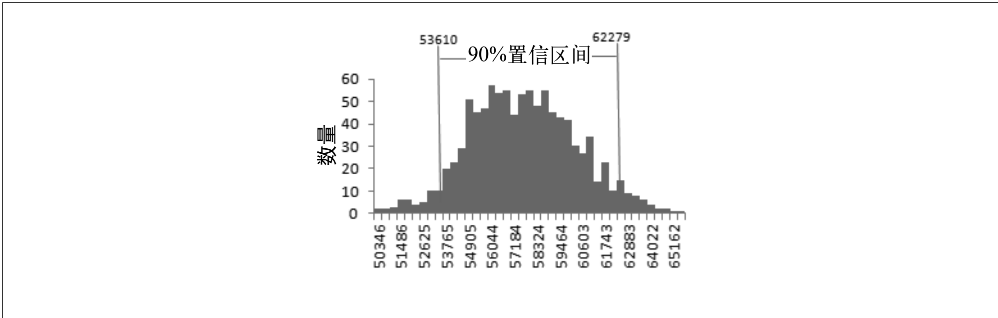


图 2-9：申请贷款者年收入均值的自助法置信区间，该区间基于规模为 20 的样本

在生成大多数统计量或模型参数的置信区间时，自助法是一种通用的工具。半个多世纪以来，统计学教材和软件一直都基于缺少计算机的统计分析，它们会使用由一些公式（尤其是 t 分布，参见 2.8 节）生成的置信区间。



当然，在得到抽样结果后，我们真正感兴趣的是“真实值落在某个特定区间中的概率是多少”。这并非置信区间真正要回答的问题，但最终是大部分人解释答案的方式。

与置信区间相关的概率问题，一开始是这样表述的：“给定抽样方法和总体，某事发生的概率是多少？”换一个角度表述就是：“给定一个抽样结果，那么某事（对总体为真的事情）发生的概率是多少？”这一问题涉及更复杂的计算，并且更难以做出估计。

置信区间所给出的百分比被称为**置信水平**。置信水平越高，置信区间越宽。此外，样本规模越小，置信区间也会越宽，即不确定性越大。两者都成立。如果要在数据更少的情况下增加置信度，那么我们必须让置信区间足够宽，以确保捕获真实值。



对于数据科学家而言，置信区间是一种了解样本结果可能的变化情况的工具。数据科学家使用这一信息时，既不是像研究人员那样为了发表学术论文，也不是为了向监管机构提交结果，而是想了解某个估计量的潜在误差情况，并确定是否需要更大的样本。

本节要点

- 置信区间是一种以区间范围表示估计量的常用方法。
- 数据越多，样本估计量的变异性越小。
- 所能容忍的置信水平越低，置信区间就越狭小。
- 自助法是一种构建置信区间的有效方法。

拓展阅读

- 用于确定置信区间的自助法，可参见 Peter Bruce 撰写的 *Introductory Statistics and Analytics: A Resampling Perspective* 一书，或是 Robin Lock 及其他四位洛克家族成员合著的 *Statistics: Unlocking the Power of Data* 一书。
- 相对于其他学科而言，需要了解测量精确度的工程师会更多地使用置信区间。Tom Ryan 撰写的 *Modern Engineering Statistics* 一书中介绍了置信区间。该书还介绍了另一种非常有用但很少被人关注的工具：预测区间。预测区间不同于均值等汇总统计量，它给出了围绕单个值的区间情况。

2.6 正态分布

呈钟形的正态分布是传统统计学中的一个标志性概念[5]。事实上，由于样本统计量的分布通常呈现出正态分布的形状，这使得正态分布业已成为一种推导样本统计量近似分布的数学公式的强大工具。

主要术语

误差

数据点与预测值或均值间的差异。

标准化

数据值减去均值，再除以标准偏差。

***z* 分数**

单个数据点标准化的结果。

标准正态分布

均值为 0、标准偏差为 1 的正态分布。

QQ 图

对样本分布与正态分布间接近程度的可视化绘图。

注 5：钟形曲线的代表性可能被高估了。美国曼荷莲学院的统计学家乔治 • 科布（George W. Cobb）在 2015 年 11 月的《美国统计学家》社论中指出："标准的统计学导论课程完全围绕正态分布展开，这超出了正态分布中心地位的实用性。"科布因其统计学导论课程的教学理念而知名。

在正态分布（如图 2-10 所示）中，68% 的数据位于均值的一个标准偏差之内，95% 的数据落于两倍的标准偏差之内。



对正态分布的一个常见误解是，该分布之所以被称为“正态分布”，是因为其中大部分数据符合正态分布，即数据值是正态的。然而，数据科学项目中使用的大部分变量（事实上，大多数原始数据）通常并不是正态分布的（参见 2.6 节）。正态分布源于很多统计量在抽样分布中是正态分布的。即便如此，只有在经验概率分布或自助法分布不可用时，才会使用正态性假设作为最后一招。

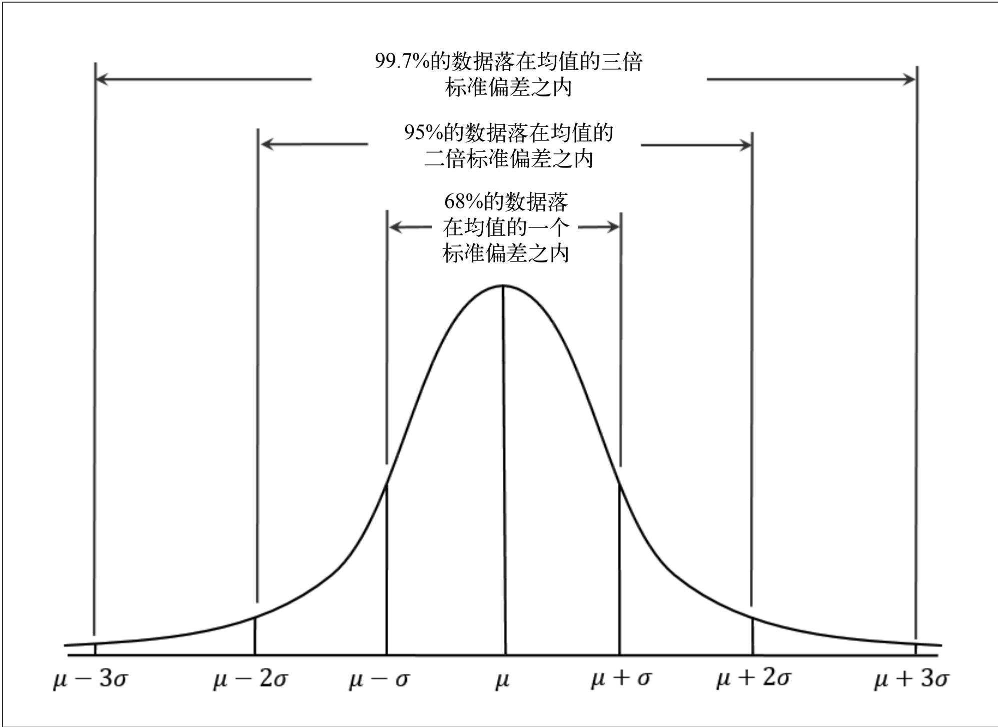


图 2-10：正态分布曲线



正态分布也被称为**高斯分布**，命名源于 18 世纪末 19 世纪初伟大的德国数学家卡尔·弗里德里希·高斯（Carl Friedrich Gauss）。正态分布还曾使用过“误差分布”这一名称。从统计学角度看，**误差**表示实际值与样本均值等统计学估计量间的差异。例如，标准偏差（参见 1.4 节）基于真实值与均值间的误差。高斯对正态分布的贡献来自于他对天体测量误差的研究，这一误差已被证明是符合正态分布的。

标准正态分布和QQ图

在**标准正态分布**中，*x* 轴的单位为距离均值的标准偏差。为了使数据能够与标准正态分布做对比，我们需要将数据值减去均值，然后除以标准偏差。这一过程被称为**归一化**或**标准化**（参见 6.1.4 节）。注意，这里所说的“标准化”与数据库记录的标准化（即转换为通用格式）无关。我们称转化值为 ***z* 分数**，正态分布有时也被称为 ***z* 分布**。

QQ 图用于可视化地确定样本与正态分布间的近似度。QQ 图对 *z* 分数从低到高进行排序，并将每个值的 *z* 分数绘制在 *y* 轴上。*x* 轴的单位是该值**秩**（rank）的正态分布所对应的分位数。由于数据是归一化的，所以单位的个数对应于数据值与均值间的距离是标准偏差的多少倍。如果数据点大体落在对角线上，那么可以近似地认为样本分布符合正态分布。图 2-11 显示了从正态分布随机生成的具有 100 个值的样本的 QQ 图。正如我们所期待的那样，数据点十分接近对角线。该图可用 R 语言的 `qqnorm` 函数生成。

```
norm_samp <- rnorm(100)
qqnorm(norm_samp)
abline(a=0, b=1, col='grey')')
```

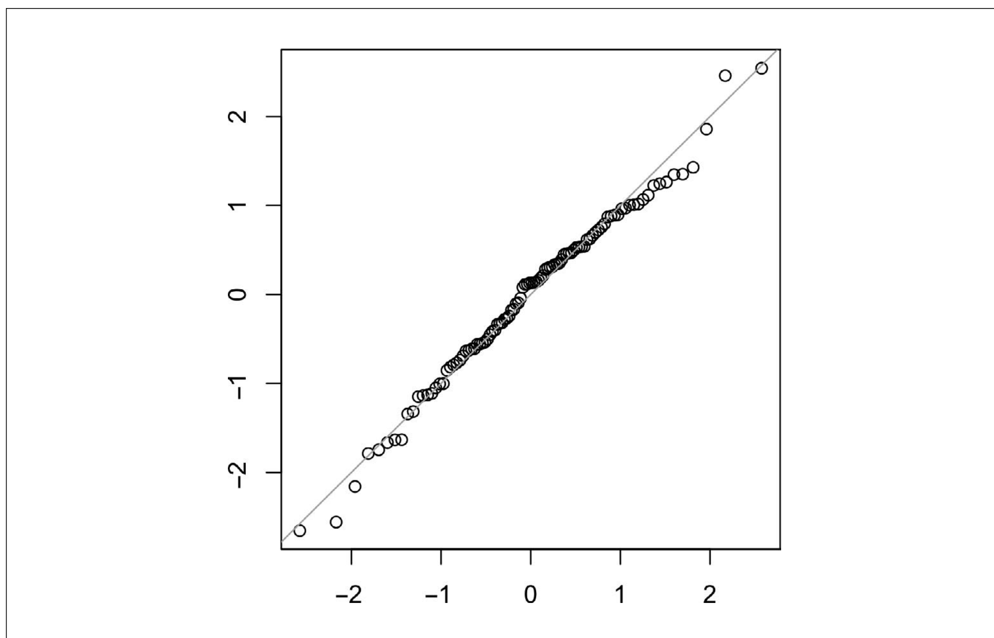
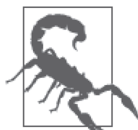


图 2-11：从正态分布随机生成的具有 100 个值的样本的 QQ 图



将数据转换为 *z* 分数（即标准化或归一化数据），并不会使数据符合正态分布。它只是将数据转化到与标准正态分布相同的尺度上，通常是为了对比。

本节要点

- 在统计学的发展史中，正态分布有着十分重要的地位，因为它允许从数学上近似不确定性和变异性。
- 虽然原始数据通常并不符合正态分布，但误差通常是符合正态分布的。对于大规模样本的均值和总数，也是一样的。
- 要将数据转换为 z 分数，需要减去数据的均值，再除以标准偏差。这样，所生成的数据才可以与正态分布进行对比。

2.7 长尾分布

尽管正态分布在统计学历史中具有非常重要的地位，但是数据通常并不符合正态分布，这与其名称完全不符。

主要术语

尾

一个频数分布的狭长部分，其中相对极值出现的频数很低。

偏斜

分布的一个尾部长于另一个尾部。

虽然正态分布非常适用于误差和样本统计量的分布，也非常有用，但是它并未表示出原始数据的分布特性。有时，数据的分布是高度**偏斜**（即不对称）的，如借款者的收入数据。有时，数据也会是离散的，如二项分布数据。对称分布和不对称分布都可能具有**长尾效应**。数据分布的尾部，对应于数据中的极值，包括极大值和极小值。在实际工作中，长尾问题（以及如何避免出现长尾问题）备受关注。纳西姆·塔勒布（Nassim Taleb）提出了**黑天鹅**理论，该理论预测异常事件（如股市崩盘）发生的可能性远大于正态分布的预测。

股票收益很好地展示了数据的长尾本质。图 2-12 显示了 Netflix 股票（NFLX）日收益情况的 QQ 图。绘图使用下面的 R 语句生成。

```
nflx <- sp500_px[,'NFLX']
nflx <- diff(log(nflx[nflx>0]))
qqnorm(nflx)
abline(a=0, b=1, col='grey')
```

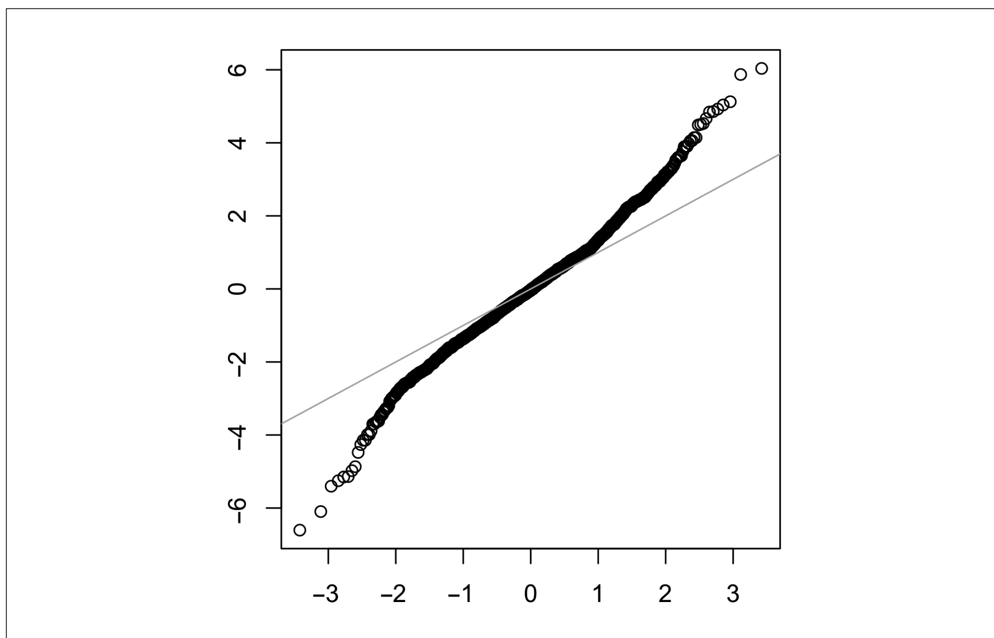


图 2-12：NFLX 股票日收益的 QQ 图

与图 2-11 不同，图 2-12 中数据点的低值远低于对角线，而高值远高于对角线。这意味着，相比于我们期望数据符合正态分布的情况，我们更趋向于观测到一些极值。图 2-12 还显示了另一种常见现象，即数据点的分布接近由落在一倍均值标准偏差范围内的数据所构成的线条。约翰·图基将此现象称为数据“在中部是正态的”，但是具有更长的尾部[6]。



大量的统计学文献研究了统计分布如何拟合观测数据的问题。我们应谨慎地使用以数据为中心的方法，它们不仅涉及科学，同样具有艺术性。从表面上看，数据是变化的，但也具有一致性。数据的分布可能具有多种形状和类型。在对给定情况建模时，通常必须借助于一些领域知识和统计学知识，才能确定适合的分布类型。例如，使用每五秒内服务器因特网流量的连续观测数据，有助于确定对“每个时间间隔的事件”建模的最优分布是否符合泊松分布（参见 2.10.1 节）。

本节要点

- 大部分数据是不符合正态分布的。
- 假设数据符合正态分布，这可导致对极端事件产生错误的估计（即“黑天鹅”现象）。

注 6：Tukey, John W. Edited by Jones, L. V. *The collected works of John W. Tukey: Philosophy and Principles of Data Analysis 1965–1986*, Volume IV. Chapman and Hall/CRC (1987). ISBN: 978-0-534-05101-3.

拓展阅读

- Nassim Taleb 撰写的《黑天鹅：如何应对不可预知的未来》。
- K. Krishnamoorthy 撰写的 *Handbook of Statistical Distributions With Applications*。

2.8 学生*t*分布

***t* 分布**呈正态分布形状，但是钟形稍厚，尾部略长。*t* 分布广泛用于描述样本统计量的分布。样本均值的分布通常呈 *t* 分布形状。*t* 分布是一个分布家族，家族中的每个成员根据样本规模的不同而有所不同。样本的规模越大，*t* 分布就越趋向于正态分布形状。

> **主要术语**
>
> *n*
>
> 表示一个样本的规模。
>
> **自由度**
>
> 自由度是一个参数，允许根据不同的样本规模、统计量和组数对 *t* 分布进行调整。

t 分布通常被称为**学生 *t* 分布**，因为它是 1908 年由格赛特（Gossett）以“学生”（Student）为作者名发表在期刊 *Biometrika* 上的。当时格赛特的雇主吉尼斯啤酒厂不想让竞争者知道自己使用了统计学方法，因此坚持要求格赛特匿名发表该论文。

格赛特在该论文中想要回答的问题是：“如果从一个大规模总体中抽取一个样本，那么样本均值的抽样分布是什么？”他从重抽样实验着手，在一个包括 3000 名罪犯的身高和左手中指长度的观测数据集中，随机地抽取了 4 个样本。（该研究属于优生学领域，所使用的是犯罪数据，关注的是发现犯罪倾向与罪犯身体或精神属性间的关联关系。）他在 *x* 轴上绘制了标准化后的结果（即 *z* 分数），在 *y* 轴上绘制了频数。由此得到了一个他称为“学生 *t*”的函数，并将该函数与样本结果拟合，绘制了对比的情况，如图 2-13 所示。

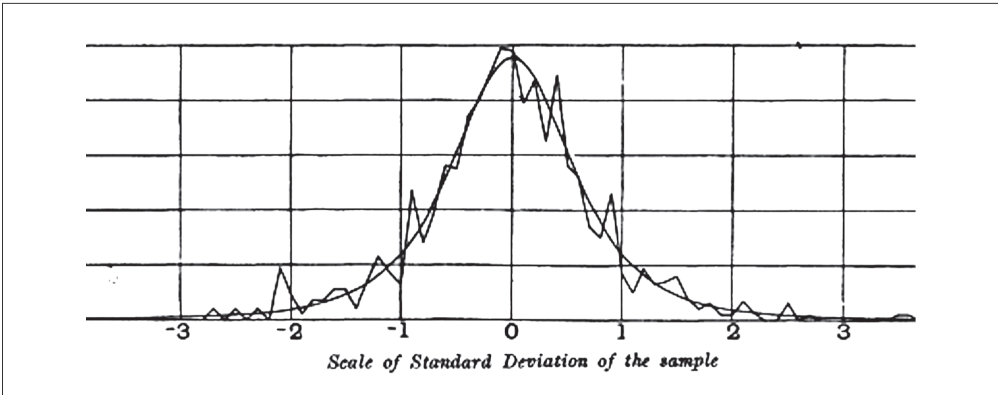


图 2-13：格赛特重抽样实验的结果，以及所拟合的 *t* 曲线（引用自他于 1908 年发表在 *Biometrika* 期刊上的文章）

我们可以将一组不同的统计量标准化，然后与 t 分布进行对比，并根据抽样变异性估计置信区间。考虑一个规模为 n 的样本，其中样本均值 $\bar{x}$ 已经计算出来。如果 s 是样本的标准偏差，那么样本均值周边 90% 的置信区间由下式给出。

$$\bar{x} \pm t_{n-1}(0.05) \times \frac{s}{n}$$

其中，$t_{n-1}(0.05)$是自由度（参见 3.7 节）为 $(n-1)$ 情况下的 t 统计量值，它表示在 t 分布的两端分别“截去”了 5%。t 分布能为样本均值的分布、两个样本均值间的差异、回归参数等统计量提供参考。

如果早在 1908 年计算能力就广泛可用，那么毫无疑问，统计量的计算从一开始就会更依赖于计算密集的重抽样方法。由于当时没有计算机，统计学家转而采用数学和函数方法，例如使用 t 分布去近似抽样分布。虽然到了 20 世纪 80 年代，计算能力的发展使得重抽样实验得以实际开展，但是教科书和软件中依然使用 t 分布及类似的分布。

要让 t 分布准确地解释样本统计量的特性，需要样本统计量的分布形状类似于正态分布。t 分布之所以被广泛使用是基于这样一个事实：即便底层的总体数据并不符合正态分布，但样本统计量通常符合正态分布。该现象被称为**中心极限定理**（参见 2.3.1 节）。



数据科学家需要了解 t 分布和中心极限定理中的哪些内容？答案是并不需要了解太多。这些分布是用于经典的统计推理中的，在数据科学中并非十分重要。对于数据科学家而言，重在理解并定量分析不确定性和变异性。这时，以实验为依据的自助抽样可以解答大多数与抽样误差相关的问题。但是，数据科学家还是时常会在统计学软件和 R 的统计过程中遇到 t 统计量，比如在 A/B 测试和回归中。因此，了解这些分布的目的对于数据科学家来说也是有所裨益的。

本节要点

- t 分布实际上是一个分布家族。它们与正态分布相似，但是尾部略厚。
- t 分布被广泛地用作样本均值分布、两个样本均值间的差异、回归参数等的参考基础。

拓展阅读

- 格赛特 1908 年发表在 *Biometrika* 期刊上的原始论文，网上提供了 PDF 文件下载（http://seismo.berkeley.edu/~kirchner/eps_120/Odds_n_ends/Students_original_paper.pdf）。
- 在大卫・莱恩提供的在线资源（http://onlinestatbook.com/2/estimation/t_distribution.html）中，可以看到对 t 分布的标准处理方法。

2.9 二项分布

> **主要术语**
>
> **试验**
>
> 一次输出离散值的事件，例如，一次硬币抛掷。
>
> **成功**
>
> 一次试验的输出为我们感兴趣的结果。
>
> **同义词**：1（相对于 0）
>
> **二项**
>
> 具有两个输出
>
> **同义词**：是 / 否、0/1、二元
>
> **二项试验**
>
> 有两种输出的试验。
>
> **同义词**：伯努利试验
>
> **二项分布**
>
> 在多次试验中（例如 *x* 次），成功次数的分布。
>
> **同义词**：伯努利分布

“是 / 否”这样的（二项）结果是数据分析的核心，因为它通常是决策或其他过程的结果，例如，买或不买，点击或不点击，存活或死亡等。**试验**对于理解二项分布至关重要。在一组试验中，每次试验有两种可能的结果，每种结果具有一个明确的概率。

例如，抛硬币 10 次是一个包含 10 次二项试验的实验，每次试验有两种可能的结果，即正面或背面朝上（如图 2-14 所示）。“是 / 否”“0/1”这样的结果称为**二元**结果，两种结果不一定都有 50% 的概率。事实上，只要两种结果的概率之和为 1 即可。统计学中的惯例做法是，将输出为“1”的试验称为一次**成功**的结果，而且通常将“1”指派给较罕见的结果。这里使用“成功”一词，并非表示结果是我们需要的或是对我们有利的，而是表示试验的确倾向于给出我们感兴趣的结果。例如，我们可能会对预测贷款拖欠或欺诈性交易感兴趣，这些事件是相对罕见的，因此我们可以将这类事件定义为“1”或“成功”。



图 2-14：北美野牛镍币的反面

二项分布是在给定每次试验的成功概率 p、实验次数 n 的情况下，成功数 x 的频数分布。根据 x、n 和 p 值的不同，二项分布也构成了一个分布家族。二项分布可以回答如下问题。

> 如果链接点击转换为购买的概率是 0.02，那么观测到 200 次点击但没有购买的概率是多少？

R 语言的 `dbinom` 函数可用于计算二项概率。

```
dbinom(x=2, n=5, p=0.1)
```

该命令返回 0.0729。该值表示每次试验成功的概率 $p = 0.1$ 时，在 $n = 5$ 次试验中观测到 $x = 2$ 的概率。

通常，我们感兴趣的是确定 x 的概率，或者 n 次实验中较罕见事件的成功概率。在这种情况下，应该使用 R 语言的 `pbinom` 函数。

```
pbinom(2, 5, 0.1)
```

该命令返回 0.9914，即在 5 次成功概率是 0.1 的试验中，观测到不多于两次成功的概率。

二项分布的均值是 $n \times p$，也可以将均值视为 n 次试验的期望成功次数，其中每次试验的成功概率是 p。

二项分布的方差是 $n \times p(1–p)$。如果试验的次数足够多（尤其是当 p 接近于 0.5 时），二项分布几乎等于正态分布。事实上，计算大规模样本的二项概率对计算能力的要求很高，因此大多数统计程序会使用具有一定均值和方差的正态分布给出近似计算。

本节要点

- 二项输出在建模中十分重要，因为它们表示了基本的决策情况，例如是否购买、是否点击、存活还是死亡等。
- 二项试验是一种具有两种可能结果的试验，其中一种结果的概率为 p，另一种结果的概率为 $1–p$。
- 当 n 很大并且 p 不接近于 0（或 1）时，二项分布可使用正态分布近似。

拓展阅读

- 阅读一个名为 quincunx 的在线弹珠模拟程序，它展示了二项分布，网址是 https://www.mathsisfun.com/data/quincunx.html。
- 二项分布是统计学导论中的主要内容。在所有的统计学导论教材中，都会有一到两章的内容介绍二项分布。

2.10 泊松分布及其相关分布

一些过程是根据一个给定的整体速率随机生成事件的。所生成的事件可能是随时间扩展的，例如某个网站的访问者情况、一个收费站的汽车到达情况等；也可能是散布于空间中

的，例如每平方米纺织品上的缺陷情况、每百行代码中的拼写错误情况。

主要术语

lambda
　　单位时间内或单位空间中的事件发生率。

泊松分布
　　单位时间内或单位空间中事件数量的频数分布。

指数分布
　　在时间或距离上，从一个事件到下一个事件的频数分布。

韦伯分布
　　泛化版本的指数分布。韦伯分布允许事件发生的速率随时间变化。

2.10.1　泊松分布

我们可以根据先验数据估计单位时间内或单位空间中的平均事件数量。但是我们可能也想知道，单位时间或单位空间之间事件的差异情况。泊松分布通过对很多单位抽样，可以告诉我们单位时间内或单位空间中事件的分布情况。例如，对于回答排队问题，泊松分布就非常有用，比如："如果要确保在 95% 的情况下，可以完全处理任意五秒内到达服务器的网络流量，我们需要多大的容量？"

泊松分布的一个关键参数是 λ（lambda）。它表示在指定时间或空间间隔中，事件发生数量的均值。泊松分布的方差也是 λ。

在模拟排队问题中使用泊松分布生成随机数，这是一种常用的方法。R 语言的 `rpois` 函数实现了此功能。该函数可以只接收两个参数，即随机数的数量和 `lambda`。

```
rpois(100, lambda = 2)
```

上面的命令将从 $\lambda = 2$ 的泊松分布生成 100 个随机数。例如，如果平均每分钟有两次客户服务电话呼入，那么上面的命令可以模拟 100 分钟内电话呼入的情况，并返回每一分钟内的电话呼入次数。

2.10.2　指数分布

指数分布可以建模各次事件之间的时间分布情况，例如，网站访问的时间间隔，汽车抵达收费站的时间间隔。它所使用的参数 λ 与泊松分布一样。在工程领域，指数分布可用于故障时间的建模；在过程管理领域，指数分布可用于对每次服务电话所需的时间进行建模。使用 R 代码实现从指数分布生成随机数时，需指定两个参数，分别是生成随机数的数量 `n` 和每个时间周期内的事件数量 `rate`。例如：

```
rexp(n = 100, rate = .2)
```

上面的代码使用每个时间周期内事件数量的均值为 2 的指数分布，生成 100 个随机数。它可用于模拟平均每分钟呼入电话 0.2 次的情况下，100 次电话的时间间隔情况（单位为分钟）。

在针对泊松分布或指数分布的模拟研究中，一个关键假设是速率 λ 在所考虑的时间周期内是保持不变的。从总体上看，这一假设很少是合理的。例如，公路或数据网络上的流量会随一天中的不同时段或者一周中的不同日子而变化。但是，我们可以将时间或空间切分为几乎同等的几个部分，这样就可以在其中做分析或模拟。

2.10.3　故障率估计

在许多应用中，事件发生率 λ 是已知的，或者可以从先验数据中估计出来。但是对于极少发生的事件，却未必如此。例如，飞机引擎发生故障就十分罕见，所以对于指定的引擎类型，几乎没有数据可用于估计发生故障的时间间隔。如果完全没有数据，就几乎没有什么基础可供估计事件的发生率。然而，我们可以做一些猜测：假如经过 20 个小时后没有发生事件，那么就可以确定事件的发生率不会是每小时一次。我们可以通过模拟或者直接计算概率，评估不同的假设事件发生率，并估计出一个阈值（发生率不可能比它低）。如果我们有一些数据，但是这些数据不足以对事件发生率做出准确可靠的估计，那么这时可以应用"拟合度检验"（参见 3.9 节）检测各种发生率，以确定它们对观察数据的拟合情况。

2.10.4　韦伯分布

在某些情况下，事件发生率并不能随时间的变化而保持恒定。如果事件的变化周期远大于事件在一般情况下的发生间隔，并没有问题。正如 2.10.2 节中所介绍的，我们只需将分析切分为多个间隔段，保持每段中的事件发生率相对固定即可。但是，如果事件发生率在每个间隔中也会发生变化，那么指数分布或泊松分布就不再有用了。在机械故障问题中，机器发生故障的风险会随时间的增加而增大，这时可能就会出现这种情况。**韦伯分布**是指数分布的一种延伸，它通过指定**形状参数** β，允许事件发生率产生变化。如果 $\beta>1$，那么事件发生率会随时间增大；如果 $\beta<1$，那么事件发生率会随时间降低。由于我们使用韦伯分布分析的是发生故障的时间，而非事件发生率，因此分布的第二个参数表示的是特征生命，而非每个时间间隔中的事件发生率。该参数也被称为**比例参数**，用 η 表示。

在使用韦伯分布时，需要估计 β 和 η 这两个参数。我们可以使用软件对数据建模，生成韦伯分布的最优拟合估计。

在使用 R 代码使用韦伯分布生成随机数时，需要指定三个参数，即生成随机数的数量 `n`、形状参数 `shape` 和比例参数 `scale`。例如，下面的代码使用形状参数为 1.5、特征生命为 5000 的韦伯分布，生成 100 个随机数字（即寿命）：

```
rweibull(100,1.5,5000)
```

本节要点

- 如果事件发生率为常数，那么可以用泊松分布对单位时间或空间内的事件数量进行建模。
- 在这种场景下，可以用指数分布对两个事件间的时间间隔或距离建模。
- 如果事件发生率会随时间变化（例如，设备故障率的增大），可以使用韦伯分布建模。

2.10.5 拓展阅读

- 在 Tom Ryan 撰写的 *Modern Engineering Statistics* 一书中，有一章专门介绍了工程应用中使用的概率分布。
- 阅读论文“Predicting Equipment Failures Using Weibu Analysis and SAS Software”和“Estimation the System Reliability Using Weibull Distribution”。这两篇论文主要从工程的角度介绍了韦伯分布的使用情况。

2.11 小结

在大数据时代，如果需要给出准确的估计量，那么随机抽样原则依然十分重要。与使用便利可用的数据相比，随机抽样可以减小偏差，并生成高质量的数据集。我们应了解各种抽样和数据生成的分布，这样才能对估计量中由随机变异性所导致的潜在误差进行量化。此外，还应了解自助法是对观测数据做有放回的抽样。对于确定样本估计量中可能存在的误差，自助法是一种“万能”的方法，颇具吸引力。

第3章

统计实验与显著性检验

实验设计是统计学实践的基石，几乎所有的研究领域都要用到实验。实验设计的目标是设计出能确认或推翻某个假设的实验。数据科学家需要开展连续的实验，尤其是与用户界面和产品营销相关的实验。本章概述了传统的实验设计方法，并指出了数据科学中常见的挑战。本章还将介绍一些在统计推断中常用的概念，并解释它们的意义以及与数据科学的相关性。

如果看到**统计显著性**、***t* 检验**或 ***p* 值**等概念，这一般是在经典统计推断"流水线"的场景下（如图 3-1 所示）。统计推断过程开始于某个假设，例如，"药物 A 要好于现有的标准药物""价格 A 比现有的价格 B 更有利可图"。实验（例如 A/B 测试）是设计用于验证假设的，我们希望所设计的实验能得出结论性的结果。实验中会收集并分析数据，进而得出结论。**推断**（inference）一词反映了这样一个意图：将从有限数据集上得到的实验结果应用于更大的过程或总体。

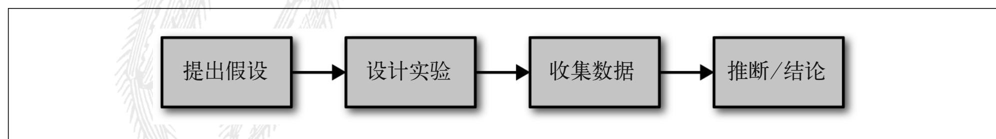


图 3-1：经典的统计推断流水线

3.1　A/B测试

A/B 测试将实验分成两个组开展，进而确定两种处理、产品、过程等中较优的一个。在两组实验中，一般会有一组采用现有的标准处理，或者是不执行任何处理，称为**对照组**，而另一组称为**实验组**。实验中的一个典型假设是实验组要优于对照组。

主要术语

处理

实验对象所接触的东西，例如药品、价格、Web 标题等。

实验组

执行特定处理的一组对象。

对照组

执行标准处理或不执行处理的一组对象。

随机化

随机地分配实验对象以进行处理的过程。

实验对象

接受处理者，例如 Web 访问者、病人等。

检验统计量

用于检验处理效果的度量。

A/B 测试的结果易于测量，因此被广泛地用于 Web 设计和营销中。下面列出了一些使用 A/B 测试的例子。

- 测试两种土壤处理，以确定哪种土壤更适合育种。
- 测试两种疗法，以确定哪种疗法对于抑制癌症更有效。
- 测试两种价格，以确定哪种价格的净利润更高。
- 测试两个 Web 标题，以确定哪个标题会带来更多的点击量（如图 3-2 所示）。
- 测试两条网络广告，以确定哪条广告能转化为更多的购买行为。

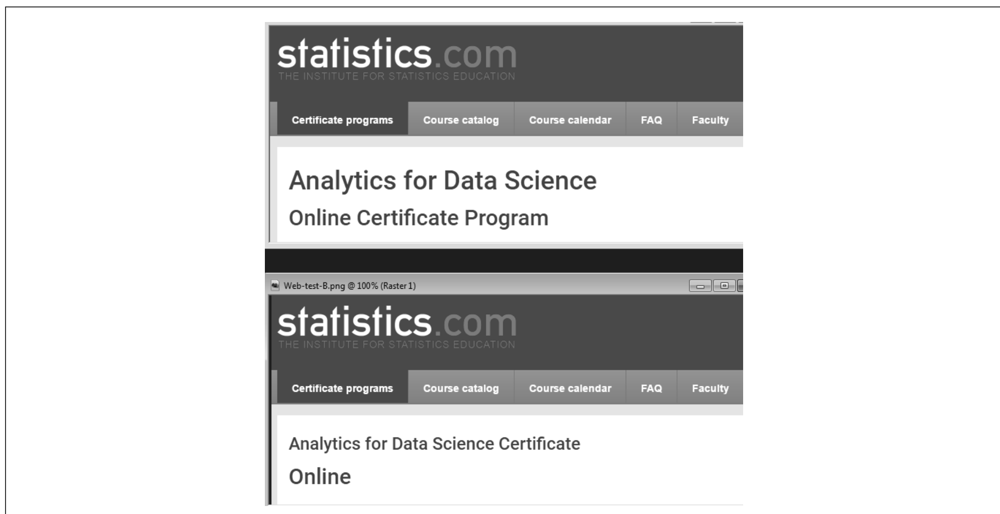


图 3-2：市场营销人员对两种 Web 展示持续进行对比测试

A/B 测试有**对象**，对象会分配给两组处理中的一组。对象可能是一个人、一种植物种子或一位 Web 访问者。注意，对象需要接受处理。在理想情况下，对象是**随机指定**（即随机分配）给一组处理的。这样，两个实验组之间的任何差异，只可能是由下面两个因素所导致的。

- 不同处理的效果。
- 将对象分配到不同处理过程中的运气因素。一些对象的效果本来就很好，而随机分配可能会导致效果好的对象集中在处理 A 或处理 B 中。

在 A/B 测试中，我们也需要关注比较 A 组和 B 组时使用的**检验统计量**（即度量）。在数据科学中，最常用的度量可能是二元变量，例如点击或未点击，购买或不购买，作弊或未作弊，等等。我们可以将比较结果归纳为一个 2×2 的表格。表 3-1 给出了实际价格测试结果的一个 2×2 的表格。

表3-1：电子商务实验结果的2×2表格

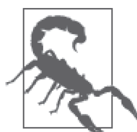
结果	价格A	价格B
点击转化为购买	200	182
点击没有转化为购买	23 539	22 406

在比较中所使用的度量，可以是连续变量（例如购买数量、利润等），也可以是计数（例如住院天数、访问的页面数量）。两者的结果显示存在着差异。如果关注的并非点击广告的转化情况，而是每次页面浏览的收益情况，那么在常见的软件输出中，表 3-1 的价格测试结果大致如下。

Revenue/page-view with price A: mean = 3.87, SD = 51.10

Revenue/page-view with price B: mean = 4.11, SD = 62.98

结果中的 SD 表示的是各组中值的标准偏差。



一些统计软件（包括 R 在内）会默认生成输出，但不能因此认为所有的输出信息都是有用的，或是与结果相关的。我们可以看到，上面给出的标准偏差就并非十分有用。它们表明数据中可能存在大量的负值，但我们知道，收入值是不可能为负的。这是由于数据集由少量较大的值（表示点击转化为购买）以及大量的零值（表示点击未转化为购买）组成。此类数据的变异性很难用单个数字总结。相对于标准偏差而言，更合理的度量是**偏离均值的绝对偏差均值**（A 组是 7.68，B 组是 8.15）。

3.1.1 为什么要有对照组

在实验中为什么不能抛开对照组，只对一个组应用我们所关注的处理，并将结果与先前的经验进行对比？

这是因为，如果没有对照组，就不能确保“其他条件均一样”，也不能确保所有差异的确是由处理（或偶然）导致的。除了处理，对照组与处理组具有相同的条件。如果我们只对比“基线”或先前的经验，那么除了处理，可能还有其他一些因素存在差异。



单盲研究和双盲研究

在**单盲研究**中，实验对象本身并不知道自己接受的是处理 A 还是处理 B。如果对象知道自身所接受的处理，那么会对响应产生影响。**双盲研究**是指研究者和协助者（例如医学研究中的医生和护士）都不知道哪个对象接受了哪种处理。如果处理是公开透明的，则盲测是不可行的，例如电脑与心理学家间的认知疗法。

在数据科学中，A/B 测试常用于 Web 领域，其中的处理可以是网页的设计、产品的价格、标题用语等。我们需要思考如何遵守随机化的原则。通常，实验对象是 Web 访问者，所关注的结果是点击数、购买、访问时长、访问的页面数量、某个页面是否被访问等。在标准的 A/B 测试中，需要预先确定一个度量。我们可能需要采集多种行为度量，而且这些度量可能是我们所关注的。但是，如果如果希望实验能在处理 A 和处理 B 这两者之间做出一个决策，那么就需要预先确立一个度量，即**检验统计量**。如果在实验开展后再去选择某个检验统计量，这无疑会引入研究人员的一些偏好。

3.1.2 为什么只有处理A和B，没有C、D……

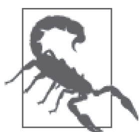
A/B 测试在营销和电子商务领域十分常见，但并非唯一的统计实验类型。我们还可以加入一些其他类型的实验，也可以对实验对象做重复的测量。例如，一些药物试验存在受试者（即实验对象）稀缺、成本高且需要不断寻找的问题，因此其实验设计中会有多个终止实验并得出结论的偶然性。

传统的统计实验设计关注的是如何回答有关具体处理的效果的静态问题。对于下面列出的两个问题，数据科学家更关注的是问题 2。

问题 1：价格 A 和价格 B 之间的差异是否是统计显著的？

问题 2：在多种可能的价格中，哪种价格最好？

鉴于此，数据科学家采用的是一种相对新颖的实验设计方法，即**多臂老虎机算法**（参见 3.10 节）。



获得实验许可

开展科学研究和医疗研究时，如果实验对象是人，那么通常需要获得他们的许可，并获得某机构审查委员会的批准。作为持续性操作的一部分的商业实验，几乎从未获得许可。大多数情况下，例如在确定价格的实验、确定显示哪个标题或是应该提供哪个报价的实验中，这种做法已被广泛接受。然而在 2014 年，Facebook 就在这个普遍接受的问题上碰了壁。该公司当时开展了一项有关用户新闻推送中的情感影响的实验。Facebook 利用情感分析技术将新闻推送帖子分为正面情感和负面情感两类，然后更改了展示给用户的内容的正负面情感的平衡。Facebook 随机选取了一些用户，向他们推送正面情感的帖子，同时向另一些用户推送负面情感的帖子。Facebook 发现，阅读正面情感帖子的用户更倾向于发布正面情感的帖子，反之亦然。尽管该实验的影响不大，但 Facebook 是在未经用户许可的情况下开展的实验，因而受到了不少批评。一些人认为，如果 Facebook 在实验中向一些极端抑郁的用户推送了负面内容的新闻，那么有可能会导致此类用户崩溃。

> **本节要点**
>
> - 将实验对象分配给两组或更多组，各组的条件完全相同，只是要接受的处理不同。
> - 在理想情况下，实验对象是随机分配给各组的。

3.1.3　拓展阅读

- 两组进行对比（即 A/B 测试）是传统统计学中一种最基本的测试。几乎任何统计学入门书都会全面地介绍 A/B 测试的设计原则和推断过程。Peter Bruce 撰写的 *Introductory Statistics and Analytics: A Resampling Perspective* 一书着重介绍了如何在数据科学场景中使用 A/B 测试和重抽样。
- 在 Web 测试中，测试的逻辑性和统计学方法同样具有挑战性。推荐从 Google Analytics 中关于实验的帮助章节入手。
- 互联网上有大量的 A/B 测试指南，其中给出的建议须谨慎对待。例如："等到大约有 1000 名访问者后，确保运行测试一周时间。"在统计学中，此类通用经验法则毫无意义。详细内容参见 3.11 节。

3.2　假设检验

假设检验也称**显著性检验**，在公开发表的传统统计学研究中随处可见。假设检验的目的是确定一个观测到的效果是否是由随机性（random chance）造成的。

> **主要术语**
>
> **零假设**
>
> 　　完全归咎于偶然性的假设。
>
> **备择假设**
>
> 　　与零假设相反，即实验者希望证实的假设。
>
> **单向检验**
>
> 　　在假设检验中，只从一个方向上计数偶然性结果。
>
> **双向检验**
>
> 　　在假设检验中，从正反两个方向上计数偶然性结果。

在构建 A/B 测试（参见 3.1 节）时，我们通常会预先构想一个假设，例如假设价格 B 可能会带来更高的利润。那么我们为什么需要做出一个假设？为什么不能只是查看实验的结果，然后选择处理结果更好的那一组？

问题的答案在于，人们在思想上倾向于低估天然随机行为的范围。一个典型的例证就是难以预料极端事件，即黑天鹅事件（参见 2.8 节）。另一个例证是人们倾向于将随机事件曲解为具有某种显著性的模式。为此，人们提出了统计假设检验方法，目的是使研究人员免受随机性的愚弄。

曲解随机性

我们可以发现，人们倾向于在实验中低估随机性。让一些朋友构想抛 50 次硬币的结果，并写下一系列随机的 H（正面朝上）和 T（反面朝上）。然后，让这些朋友实际去抛 50 次硬币，并记录结果。将真正的硬币抛掷结果和人工生成的结果各放一处。我们很容易看出哪个结果是真实的，因为真实的结果中会出现一组连续的 H 或 T。在真实的抛 50 次硬币中，常常能看见五六个连续的 H 或 T。但是，我们大多数人在构想随机抛硬币的结果时，如果已经连续有三四个 H，就会暗示自己，最好在这里就切换为 T，这样序列看上去更随机一些。

另外，抛硬币实验也说明了一个问题。如果的确在现实世界中看到了类似于连续出现 6 个 H 这样的事情，例如一个标题比另一个标题好 10%，我们倾向于将其归因于真实情况，而非巧合。

在一个设计适当的 A/B 测试中，处理 A 和处理 B 之间任何可观测到的差异，必定是由下面两个因素之一所导致的。

- 分配对象中的随机可能性
- 处理 A 和处理 B 之间的真实差异

统计假设检验是对 A/B 测试（或任何随机实验）的进一步分析，意在评估随机性是否可以合理地解释 A 组和 B 组之间观测到的差异。

3.2.1 零假设

假设检验使用的逻辑是：“鉴于人们倾向于对异常的随机行为做出反应，并将其解释为有意义的真实行为，我们要在实验中证明，组间差异要比偶然性可能导致的差异更极端。”这里包含了一个基线假设，即各个处理是等同的，并且组间差异完全是由偶然性所导致的。我们称该基线假设为**零假设**。事实上，我们希望能证明零假设是**错误的**，并证明 A 组和 B 组结果之间的差异要比偶然性可能导致的差异更大。

一种实现方式是通过重采样置换过程，对 A 组和 B 组的结果做随机混洗，并反复将数据分配为规模相近的组，之后查看实验得到的差异与观测差异同样极端的频率。更多内容参见 3.3 节。

3.2.2 备择假设

假设检验本身不仅包括零假设，还包括一个相抵消的备择假设。下面通过一些例子来说明。

- 零假设是“A 组和 B 组的均值间没有差异”，备择假设是“A 不同于 B”（可能更大，也可能更小）。
- 零假设是“A ⩽ B”，备择假设是“B > A”。
- 零假设是“B 不会比 A 大 *x*%”，备择假设是“B 比 A 大 *x*%”。

总而言之，零假设和备择假设必须涵盖了所有的可能性。假设检验的结构取决于零假设的性质。

3.2.3　单向假设检验和双向假设检验

A/B 测试通常是根据一个已有的默认选项（比如 A）去测试一个新的选项（比如 B），并且假定除非证明 B 明显优于 A，否则我们将坚持使用 A。在这种情况下，我们需要一个假设检验来免受倾向于 B 的偶然性的欺骗。我们并不在意在另一个方向上是否会受偶然性的愚弄，因为除非能证明 B 更好，否则我们将坚持 A。所以，我们需要一种**有方向的**备择假设（即 B 比 A 好）。这种情况下，我们可以使用**单向**（或“单尾”）假设检验。这意味着极端偶然性只会导致从一个方向上计入 p 值。

如果想要假设检验使我们免受任意方向上偶然性的愚弄，那么备择假设应该是**双向**的（即 A 不同于 B，它可能更大，或是更小）。在这种情况下，我们要使用**双向**（或“双尾”）假设。这意味着极端偶然性导致可以从任意一个方向上计入 p 值。

单向假设检验通常遵循 A/B 决策过程，即需要指定一个选项，并且除非证明另一个选项更好，否则将指定该选项为“默认”的。然而，包括 R 在内的一些软件的默认输出通常提供的是双向测试，并且许多统计学家为了避免争议，也会选择更为保守的双向测试。选择单向还是双向，这是一个让人困惑的问题，但是该问题与数据科学的关系并不大。在数据科学中，p 值的计算精度并非十分重要。

本节要点

- **零假设**的逻辑理念体现为没有特殊事件发生，任何观察到的效果都是由随机偶然导致的。
- **假设检验**假定零假设为真，创建“零模型”（一种概率模型），并检验所观察到的效果是否是该模型的合理结果。

3.2.4　拓展阅读

- Leonard Mlodinow 撰写的 *The Drunkard's Walk: How Randomness Rules Our Lives* 一书综述了“随机性控制我们生活”的方式。
- David Freedman、Robert Pisani 和 Roger Purves 的经典统计学教材 *Statistics*（第 4 版）。该书没有采用罗列数学理论的方式，并很好地介绍了大部分统计学内容，其中包括假设检验。
- Peter Bruce 撰写的 *Introductory Statistics and Analytics: A Resampling Perspective* 一书，从重抽样角度介绍了假设检验的概念。

3.3　重抽样

在统计学中，**重抽样**是指从观测数据中反复地抽取数据值，目标是评估一个统计量中的随机变异性。重抽样还可用于评估并提高一些机器学习模型的准确性。例如，对于使用多个自助数据集构建的决策树模型，可以通过 Bagging 过程计算其平均值，参见 6.3 节。

重抽样过程主要有两种类型，即**自助法**和**置换检验**。自助法用于评估一个估计量的可靠

性，我们在前面已经做了介绍（参见 2.4 节）。本节将介绍用于检验假设的置换检验，它通常涉及两组或多组。

主要术语

置换检验

将两组或多组样本组合在一起，并将观测值随机地（或穷尽地）重新分配给重抽样。

同义词：随机化检验、随机置换检验、准确检验

有放回，无放回

在抽样时，所抽取的元素在下一次抽取前是否放回样本中。

3.3.1 置换检验

置换过程涉及两组或多组样本，通常是 A/B 测试或其他假设检验中的组。置换意味着改变一组值的顺序。要对一个假设进行**置换检验**，首先要将从 A 组和 B 组（当然还可以包括其他组，例如 C、D……）中得到的结果组合在一起。这就是零假设的逻辑，即无论处理指定给哪个组，都是无差别的。然后，我们从组合集中随机抽取出各个组，并查看组间的差异情况，实现对假设的检验。置换过程如下。

(1) 将各个组得出的结果组合为一个数据集。
(2) 对组合得到的数据做随机混洗，然后从中随机抽取（有放回）一个规模与 A 组相同的重抽样样本。
(3) 在余下的数据中，随机抽取（无放回）一个规模与 B 组相同的重抽样样本。
(4) 如果还有 C 组、D 组甚至更多的组，执行同样的操作。
(5) 无论对原始样本计算的是哪一种统计量或估计量（例如，组比例差异），现在对重抽样进行重新计算，并记录结果。这构成了一次置换迭代。
(6) 重复上述步骤 R 次，生成检验统计量的置换分布。

现在我们回头查看所观测到的组间差异，并与置换差异进行对比。如果观测到的差异位于置换差异内，那么置换检验的结果并不能证实任何事情，因为观测到的差异落在偶然可能产生之差异的范围内。但是，如果观测到的差异大部分落在置换分布之外，那么我们就可以得出“与偶然性无关”这一结论。如果使用专业术语描述，我们称差异是**统计显著**的（参见 3.4 节）。

3.3.2 例子：Web黏性

有一家公司提供较高价格的服务。现在，该公司想要测试两种 Web 显示，以确定哪一种能带来更高的销售额。由于该公司提供的服务价格较高，因此销量并不大，而且销售周期很长。要想确定哪种 Web 显示的效果更好，该公司需要很长的时间才能积累到足够多的销售数据。鉴于此，该公司决定使用一种代理变量来度量结果，并使用详细描述公司服务的内部页面替代。



代理变量是一种可以代表我们所关注的真正变量的变量。真正关注的变量可能不可用，也可能度量的成本太高或耗时过长。例如，在气候研究中，远古冰芯的含氧量被用作温度的代理变量。最好至少有一点关于真正变量的数据，这样可以评估真正变量与代理变量间的关联程度。

在本例中，一个潜在的代理变量是着陆页上的点击数。当然，更好的代理变量是访问者在页面上停留的时间。可以认为，如果一个 Web 显示页面能吸引人们关注更长的时间，那么它就可能会带来更高的销售额。因此，我们这里所采用的度量是页面 A 与页面 B 的平均会话时间。

由于检验中所使用的 Web 页面是内部专用的，因此并不会有大量的访问者。另外应注意的是，我们使用了 Google Analytics（GA）工具测定会话时间，但是 GA 无法测定访问者上次访问的会话时间。不过，GA 并不从数据中删除该会话，而是将记录置为零，因此我们需要对数据做一些额外的处理，以从数据中删除这些会话。基于此，我们就两种不同的 Web 显示合计得到了 36 个会话，其中页面 A 的会话为 21 个，页面 B 的会话为 15 个。为了直观地比较会话时间，我们使用 ggplot 实现了箱线图的并排绘制。

```
ggplot(session_times, aes(x=Page, y=Time)) +
  geom_boxplot()
```

生成的箱线图如图 3-3 所示。图中显示了页面 B 具有比页面 A 更长的会话时间。各组的均值计算方式如下所示。

```
mean_a <- mean(session_times[session_times['Page']=='Page A', 'Time'])
mean_b <- mean(session_times[session_times['Page']=='Page B', 'Time'])
mean_b - mean_a
[1] 35.66667
```

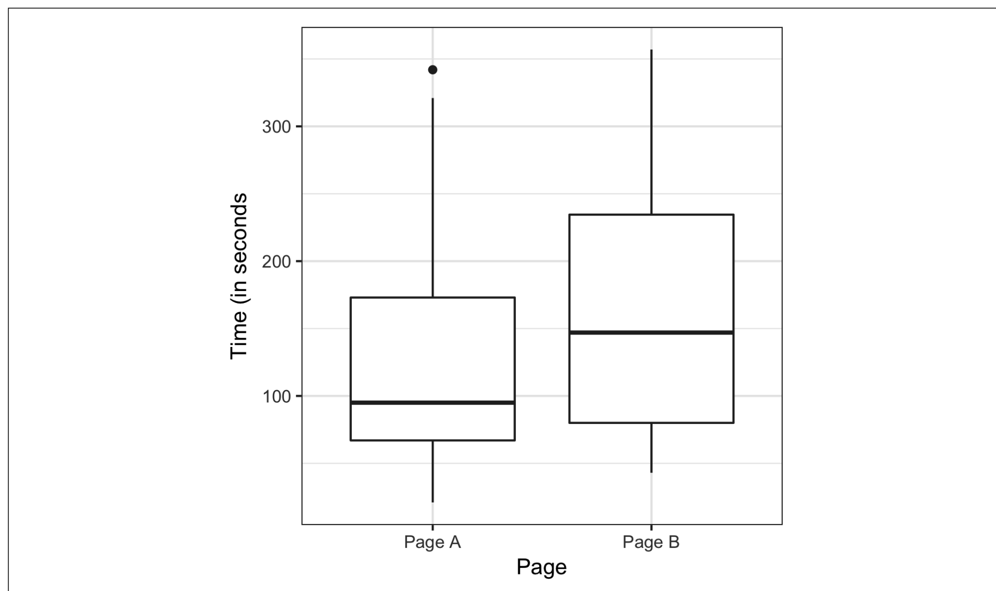


图 3-3：页面 A 和页面 B 的会话时间

页面 B 的会话时间更长，平均比页面 A 高出约 35.7 秒。但问题在于，这一差距是否落在随机性可能生成的范围内，即是否是统计显著的。要回答这一问题，一种方法是应用置换检验，将所有会话时间组合在一起，然后反复做随机混洗，再将数据分为一个具有 21 个观测值的组（页面 A，$n = 21$）和一个具有 15 个观测值的组（页面 B，$n = 15$）。

我们实现了一个进行置换检验的函数。该函数可以将 36 个会话时间随机分配给一个具有 21 个元素的组（页面 A）和一个具有 15 个元素的组（页面 B）。这个函数的代码如下。

```
perm_fun <- function(x, n1, n2)
{
  n <- n1 + n2
  idx_b <- sample(1:n, n1)
  idx_a <- setdiff(1:n, idx_b)
  mean_diff <- mean(x[idx_b]) - mean(x[idx_a])
  return(mean_diff)
}
```

该函数的工作原理是，无放回地抽样 n2 次，并分配给 B 组，余下的 n1 次抽样分配给 A 组。函数返回两组均值之间的差异。我们指定 n2 = 15，n1 = 21，并调用该函数 $R = 1000$ 次，然后绘制所生成的会话时间差异分布情况的直方图。

```
perm_diffs <- rep(0, 1000)
for(i in 1:1000)
  perm_diffs[i] = perm_fun(session_times[,'Time'], 21, 15)
hist(perm_diffs, xlab='Session time differences (in seconds)')
abline(v = mean_b - mean_a)
```

图 3-4 显示了生成的直方图。从图中可以看出，对于页面会话时间，随机置换的均值差异通常会超出观测到的差异（图中的垂直线）。这表明，页面 A 和页面 B 会话时间间的观测差异落在随机变异的范围内，因此不是统计显著的。

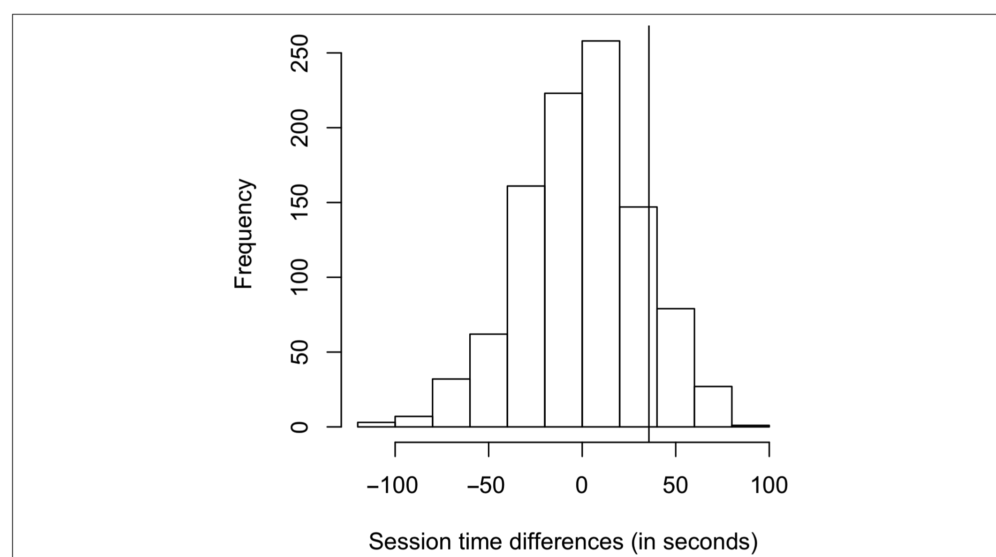


图 3-4：页面 A 和页面 B 会话时间差异的频数分布

3.3.3 穷尽置换检验和自助置换检验

置换检验除了使用前面介绍的随机混洗过程（也称**随机置换检验**或**随机检验**）之外，还有两种重要变体：

- 穷尽（exhaustive）置换检验
- 自助（bootstrap）置换检验

穷尽置换检验并不是随机混洗并分组数据，而是尝试所有可能的分组。穷尽置换检验只适用于规模较小的样本。如果做大量的重复混洗，那么随机置换检验的结果会近似于穷尽置换检验的结果，并在极限上逼近。穷尽置换检验有时也被称为**准确检验**，因为其统计学属性确保了零模型不会被检验为比 α 值水平更显著（参见 3.4 节）。

自助置换检验是在置换检验第二步和第三步的抽取中，进行有放回抽样，而非无放回抽样。这样，重抽样过程不仅建模了随机分配对象给处理的过程，而且建模了从总体中随机抽取对象的过程。这两个都是统计学过程，但是它们之间的差别过于复杂，因而不被数据科学实践所关注。

3.3.4 置换检验：数据科学的底线

在探索随机变异性中，置换检验是一种十分有用的启发式过程。它很容易编码，也很容易理解和解释。针对统计学中那些基于公式的形式主义和“假决定论”，置换检验提供了切实可行的绕行方法。

不同于依赖于统计学公式的方法，重抽样的一个优点在于给出了一种更加近乎于“万能”的推断方法。它所适用的数据可以是数值，也可以是二元的；样本规模可以相同，也可以不同；并且无须假设数据符合正态分布。

本节要点

- 置换检验将多个样本组合在一起，并做随机混洗。
- 对混洗后的值做分组并重抽样，计算我们感兴趣的统计量。
- 重复上述过程，并在表格中记录重抽样统计量的情况。
- 对比统计量的观测值与重抽样分布，就可以判定观测到的样本间差异是否由偶然性导致的。

3.3.5 拓展阅读

- Eugene Edgington 和 Patrick Onghena 合著的 *Randomization Tests*。不要过度沉溺于该书中的非随机抽样内容。
- Peter Bruce 撰写的 *Introductory Statistics and Analytics: A Resampling Perspective* 一书。

3.4 统计显著性和*p*值

统计学家引入了统计显著性的概念，用于衡量一个实验（也可以是对已有数据的研究）所

生成的结果是否会比随机情况下可能给出的结果更极端。如果生成的结果超出了随机变异的范围，则我们称它是统计显著的。

> **主要术语**
>
> **p 值**
>
> 对于一个加入了零假设的偶然性模型，p 值指得到与观测结果一样不寻常或极端的结果的概率。
>
> **α 值**
>
> 在实际结果的确是统计显著的情况下，α 值指偶然性结果必须超出的“不寻常性”概率的阈值。
>
> **第一类错误**
>
> 错误地将一个由随机导致的效果归结为真。
>
> **第二类错误**
>
> 错误地将一个为真的效果归结为由随机导致的。

以表 3-2 为例，表中数据是 3.1 节中 Web 测试的结果。

表3-2：电子商务实验结果的2 × 2表格

结果	价格A	价格B
点击转化为购买	200	182
点击没有转化为购买	23 539	22 406

价格 A 的转化情况比价格 B 好近乎 5%（0.8425% 对比 0.8057%，差异为 0.0368%）。当业务量很大时，这一差异就会具有显著的意义。一个有超过 4.5 万条数据的集合，完全可以被视为“大数据”，没有必要做统计显著性检验，统计显著性检验主要针对的是小规模样本中的抽样变异性。不过我们也能看到，此例中的转化率非常低，甚至小于 1%，以至于实际有意义的值（即转化）只有数百个。事实上，所需的样本规模取决于转化率。我们可以使用重抽样，检验价格 A 与价格 B 之间的转化差异是否位于随机变异的范围内。这里所说的**随机变异**（chance variation），是指在概率模型中加入“两者在转换率上不存在差异”这一零假设后，由模型生成的随机变异性（参见 3.2.1 节）。下面我们给出一个置换过程，该过程的目的是要回答如下问题：“如果两种价格具有相同的转换率，那么随机变异的方差能否产生 5% 的差异？”

(1) 将所有的样本结果置于同一个桶中。同一个桶表示假定两种价格具有相同的转换率。在本例中，我们有 200 + 182 = 382 个 1，23539 + 22406 = 45945 个 0，这样转换率为 382/(45945 + 382) = 0.008246 = 0.8246%。

(2) 在桶中做随机混洗，并从中抽出规模为 23 739（与价格 A 的 n 值相同）的重抽样，记录抽样中 1 的个数。

(3) 记录桶中余下 22 588（与价格 B 的 n 值相同）个数据点中 1 的个数。

(4) 记录两者中 1 的比例在百分位数上的差异。

(5) 重复第 2 步到第 4 步多次。
(6) 计算其中差异大于或等于 0.0368% 的频数。

下面，我们再次使用 3.3.2 节中定义的函数 perm_fun，创建随机置换转换率差异的直方图。

```
obs_pct_diff <- 100*(200/23739 - 182/22588)
conversion <- c(rep(0, 45945), rep(1, 382))
perm_diffs <- rep(0, 1000)
for(i in 1:1000)
  perm_diffs[i] = 100*perm_fun(conversion, 23739, 22588 )
hist(perm_diffs, xlab='Session time differences (in seconds)')
abline(v = obs_pct_diff)
```

生成的绘图如图 3-5 所示，图中显示的直方图是 1000 次重抽样的结果。在本例中，我们观察到的差异 0.0368 落在随机差异的范围内。

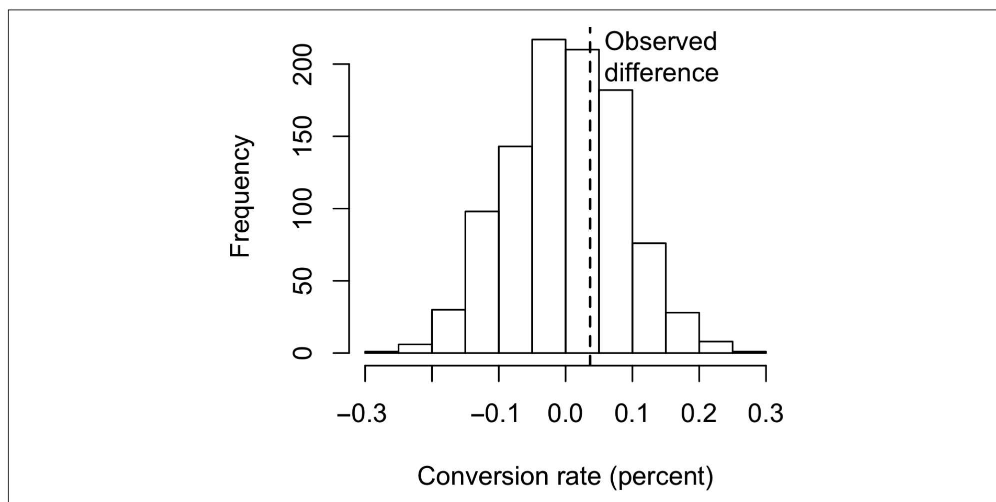


图 3-5：价格 A 和价格 B 的转换率差异的频数分布

3.4.1 *p*值

在衡量统计显著性时，简单地查看绘图并不是一个非常精确的方法，人们更关注的是 ***p* 值**。p 值表示随机模型生成的结果比观测结果更极端的频数。在估计置换检验的 p 值时，我们可以采用置换检验中生成大于或等于观测差异值的检验次数所占的比例。

```
mean(perm_diffs > obs_pct_diff)
[1] 0.308
```

结果显示 p 值为 0.308，这意味着随机性给出的差异，有望在约 30% 的情况下大于或等于观测差异[1]。

注 1：此处 p 值的计算具有随机性，因此在实际运行示例程序时，给出的 p 值可能是一个与本例输出近似的值。——译者注

在本例中，我们不需要使用置换检验也可以获得 p 值。根据二项分布，我们可以使用正态分布近似估计 p 值。在使用 R 语言编程时，函数 `prop.test` 执行该操作。

```
> prop.test(x=c(200,182), n=c(23739,22588), alternative="greater")

	2-sample test for equality of proportions with continuity correction

data: c(200, 182) out of c(23739, 22588)
X-squared = 0.14893, df = 1, p-value = 0.3498
alternative hypothesis: greater
95 percent confidence interval:
 -0.001057439  1.000000000
sample estimates:
     prop 1      prop 2
0.008424955 0.008057376
```

在函数的输出中，参数 `x` 表示各组的成功次数，参数 `n` 是试验次数。我们可以看到，由正态近似所生成的 p 值为 0.3498，接近于使用置换检验所得到的 p 值。

3.4.2　α值

完全根据研究人员的判断力去确定一个结果是否“太不寻常”因而是偶然的，统计学家无疑会对此做法大皱眉头。在统计学家看来，正确的做法是提前设定一个阈值，例如“超过随机（零假设）结果 5%”。这样的阈值被称为 α 值。α 值的常见取值是 5% 和 1%。α 值的选取具有一定的随意性——该过程无法确保在 x% 的情况下做出正确的决策。原因在于我们要回答的概率问题并不是“随机发生的概率是多少”，而是“给定一个随机模型，出现极端结果的概率是多少”。这样我们需要对随机模型的适当性进行逆推，但是在判断过程中没有任何可依据的概率。这个问题一直困扰着统计学家。

*p*值的意义

近年来，对 p 值的使用一直存在着相当大的争议。一份心理学期刊甚至“禁止”在其收到的论文中使用 p 值，理由是如果只根据 p 值做出论文可以出版的决定，那么会导致一些质量不好的研究得以发表。有太多的研究人员只是大概了解 p 值的真正含义，就根据数据和各种可能的假设开展检验，直到找出一种可以生成显著 p 值的组合，由此撰写出一篇适合发表的论文。

真正的问题在于，我们希望 p 值能包含更多的意义，并且希望 p 值能够表达如下信息。

> 结果由随机所导致的概率。

而且我们希望该值越低越好，这样就可以得出某一假设得到证明的结论。这也是不少期刊编辑对 p 值的解释。但 p 值实际所表示的是如下含义。

> **给定一个随机模型**，模型所给出的结果与观测结果同样极端的概率。

这两者之间的差异并不明显，但的确存在。显著的 p 值并非如看上去那样，它并不能引导我们沿着一条似乎正确的“证明”道路走下去。如果我们理解了 p 值的真正含义，那么就此得出“统计显著”结论的逻辑基础是不太稳固的。

2016 年 3 月，美国统计协会（ASA）在经过内部审议后，发表了一份关于 p 值使用的警告性声明，其中揭示了人们对 p 值的误解程度。

美国统计协会的声明指出了针对研究人员和期刊编辑的六项原则。

(1) p 值可以表示数据与指定统计模型间的不兼容程度。
(2) p 值并不能测量所研究的假设为真的概率，也不测量仅通过随机性生成数据的概率。
(3) 不应仅根据 p 值是否超过了给定的阈值，就得出一个科学结论，或做出一个商业或政策决定。
(4) 正确的推断需要具有全面的报告和完全的透明度。
(5) p 值（或统计显著性）并不测量效果的规模，也不测量结果的重要性。
(6) p 值本身并不能提供一种对模型或假设的很好量度。

3.4.3 第一类错误和第二类错误

在评估统计显著性时，可能会出现下面两种类型的错误。

- **第一类错误**：错误地将仅由随机性导致的效果判定为真。
- **第二类错误**：错误地将实际为真的效果判定为假的（即由随机性导致的）。

事实上，第二类错误并不是一种错误，它是由于判断样本的规模过小，而无法检测到效果。如果 p 值不足以表明统计显著性（例如，超过 5%），我们应称其为“效果未验证”。增大样本的规模，可能会生成较小的 p 值。

显著性检验（即**假设检验**）的基本功能就是防止我们被随机性愚弄。因此，我们通常可以通过构造显著性检验去最小化第一类错误。

3.4.4 数据科学与 p 值

数据科学家所做的工作一般并不会发表在科学期刊上，因此对 p 值意义的辩论是颇具学术性的。如果数据科学家想知道一个看上去有意义并且有用的模型结果是否落在随机变异的范围内，p 值是一种有用的指标。作为一种在实验中使用的决策工具，p 值不应被视为一种决定性的因素，而应被视为是另一种可以辅助决策的信息。例如，有时可以将 p 值作为一些统计学或机器学习模型的中间输入值，根据 p 值决定一个特征应该包含在模型中，还是应该从模型中排除。

本节要点

- 显著性检验可以用于确定观测到的效果是否落在零假设模型的随机变异范围内。
- 给定一个零假设模型，p 值表示模型所生成的结果与观测到的结果同样极端的概率。
- α 值是零假设随机模型“不寻常性”的阈值。
- 相对于数据科学而言，显著性检验在正式的研究报告中更加重要。但是近年来，即便是对于研究报告，p 值的重要性也一直在下降。

3.4.5 拓展阅读

- Stephen Stigler 的论文“Fisher and the 5% Level”对 Ronald Fisher 1925 年出版的 *Statistical Methods for Research Workers* 一书做了综述，其中重点关注了 5% 的显著性水平。
- 参见 3.2.4 节“拓展阅读”的内容。

3.5 *t*检验

显著性检验具有多种类型，具体取决于数据集是计数数据还是测量数据、所具有的样本数量以及测量的具体内容。***t* 检验**是其中一种十分常用的检验，其命名源于最初由 W. S. Gossett 提出的学生 *t* 分布。*t* 分布用于估计单个样本均值的分布情况（参见 2.8 节）。

主要术语

检验统计量
对我们所关注的差异或效果的度量。

***t* 统计量**
归一化的检验统计量。

***t* 分布**
一种用于比较所观测到的 *t* 统计量的参考分布。对于 *t* 检验，参考分布是从零假设生成的。

所有的显著性检验都要求指定一个**检验统计量**去测量所关注的效果，并确定观测到的效果是否落在随机变异的范围内。在重抽样检验（参见 3.3.1 节对置换的介绍）中，数据的规模并不是十分重要。我们从数据本身创建参考（零假设）分布，并据此使用检验统计量。

统计假设检验形成于 20 世纪 20 年代和 30 年代，当时无法做到对数据随机混洗数千次，以用于重抽样检验。但是统计学家发现，*t* 检验很好地近似了置换（随机混洗）分布。*t* 检验基于格赛特提出的 *t* 分布，可以在十分常见的两个样本的比较（即 A/B 测试）中使用，只要样本中的数据是数值型的。但是在使用 *t* 分布时，为了排除规模因素的影响，必须对检验统计量做归一化处理。

经典的统计学教材在介绍 *t* 检验时，会列出多个公式，其中包含了格赛特提出的 *t* 分布。还会介绍如何对数据做归一化，以便与标准 *t* 分布做比较。但是在本书中，我们并不会给出这些公式，因为这些公式已经包含在 R 和 Python 等统计软件的常用命令中。在 R 语言中，我们可以使用函数 `t.test`。

```
> t.test(Time ~ Page, data=session_times, alternative='less' )

        Welch Two Sample t-test

data:  Time by Page
t = -1.0983, df = 27.693, p-value = 0.1408
alternative hypothesis: true difference in means is less than 0
```

```
95 percent confidence interval:
     -Inf 19.59674
sample estimates:
mean in group Page A mean in group Page B
            126.3333             162.0000
```

其中的备择假设是页面 A 会话时间的均值小于页面 B 的。给出的 *p* 值非常接近置换检验的 *p* 值 0.124，参见 3.3.2 节。

我们可以使用重抽样，构造出一个能够反映观测数据和要检验的假设的解决方案，而无须关心数据是数值型还是二元的，样本的规模是否平衡，以及样本方差等因素。如果使用统计学公式，很多变异性可以表示为公式形式，但是这些公式可能会令人困惑。统计学家需要依靠公式去探索问题并按图索骥，但是数据科学家并不需要这样做。通常，数据科学家并不需要钻研假设检验和置信区间的细枝末节，这些是研究人员在准备论文以便展示时需要搞清楚的。

本节要点

- 在计算机出现之前，重抽样检验并不实用，统计人员使用标准参考分布。
- 检验统计量应该做归一化，这样才能与参考分布做比较。
- *t* 统计量是一种广为使用的归一化统计量。

拓展阅读

- 任何一本统计学入门教材都会介绍 *t* 统计量及其用途。在此我们推荐两本教材。一本是 David Freedman、Robert Pisani 和 Roger Purves 合著的经典统计学教材 *Statistics*（第 4 版），另一本是 David S. Moore 撰写的 *The Basic Practice of Statistics*。
- 关于 *t* 检验和重抽样过程的并行处理，推荐阅读 Peter Bruce 撰写的 *Introductory Statistics and Analytics: A Resampling Perspective* 一书，或者 Robin Lock 及其他四位洛克家族成员合著的 *Statistics: Unlocking the Power of Data* 一书。

3.6 多重检验

在 2.2 节中我们曾提及，统计学中有一句话："如果拷问数据的时间足够长，那么它迟早会招供。"这意味着，如果我们能从足够多的视角去观察数据，并提出足够多的问题，几乎总是可以发现具有统计显著性的效果。

主要术语

第一类错误

错误地得出一个效果是统计显著的结论。

错误发现率

在多重检验中，犯第一类错误的比率。

***p* 值校正**

用于在同一数据上做多重检验。

过拟合

拟合了噪声。

例如，给定随机生成的 20 个预测变量和一个结果变量，如果进行一组 20 次 $\alpha = 0.05$ 水平的显著性检验，那么很可能至少有一个预测因子会（错误地）显示为统计显著的。如上所述，这被称为**第一类错误**。在计算第一类错误的概率时，可以首先计算在 0.05 水平上所有预测因子将被正确检验为非统计显著的概率。在本例中，一个预测因子被正确地检验为非统计显著的概率是 0.95，那么全部 20 个预测因子被正确地检验为非统计显著的概率就是 $0.95 \times 0.95 \times 0.95 \times \cdots$，即 $0.95^{20} = 0.36$。[2] 至少一个预测因子将被错误地验证为显著的概率，就是 1 减去所有预测因子都是非统计显著的概率等于 0.64。

上面介绍的问题涉及数据挖掘中的过拟合问题，即“模型拟合了噪声”。如果我们添加的变量越多，或者运行的模型越多，那么偶然出现“统计显著性”的概率就会越大。

在有监督学习任务中，会给出一个验证集，让模型评估从未见过的数据，从而降低了风险。在没有已标记验证集的统计学习和机器学习任务中，仍然存在由统计噪声得出结论的风险。

统计学提供了一些过程，可以在一些特定的场景下解决这个问题。例如，在比较多个处理组的结果时，我们可以提出多个问题。例如，对于处理 A、B 和 C，我们可以提出如下问题。

- A 是否不同于 B？
- B 是否不同于 C？
- A 是否不同于 C？

另一个例子是在临床试验中，我们可能想要在多个阶段查看某种治疗的效果。在每个阶段，我们都可以提出多个问题，每个问题都会增加被随机性愚弄的可能性。为了解决这一问题，统计学给出了一种**校正**（adjustment）过程。相比于单一假设检验所设置的统计显著性界限，校正过程设置了更严格的统计显著性界限。校正过程通常涉及根据校正校正检验的次数“划分 α 值”。这导致了对每次检验使用较小的 α 值，即对于统计显著性更严格的界限。**Bonferroni 校正**就是这样的一种过程，它仅是将 α 值除以观测次数 n。

注 2：乘法规则指出，n 个独立事件一并发生的概率，是每个事件发生概率的乘积。例如，如果你我两人每人抛一次硬币，那么我们的硬币都正面向上的概率是 $0.5 \times 0.5 = 0.25$。

然而，多重比较问题超出了上面列举的这些高度结构化的案例，它与反复“数据疏浚”（dredging）现象有关。这一现象催生了“拷问数据”的谚语。也就是说，给定一组足够复杂的数据，如果你没有从中找到感兴趣的内容，那么说明你根本就没有尽力去查看数据。现在，可供使用的数据达到了前所未有的规模，在 2002 年至 2010 年期间，发表的论文数也近乎翻了一番。这为在数据中发现有意义的内容提供了很多偶然性，其中包括下列多重性问题。

- 如何两两成对地查看多个组间的差异情况。
- 对于以各种方式构建的数据子集，如何查看结果。例如，“我们并未在总体中发现显著的处理效果，但在 30 岁以下的未婚女性这一子集中，发现了显著的处理效果”。
- 如何尝试使用多种统计模型。
- 如何在模型中加入多个变量。
- 如何询问多个不同的问题，即不同的可能结果。



错误发现率

错误发现率这一术语，最初用于描述一组给定的假设检验错误地识别显著效果的比率。随着基因组研究的发展，错误发现率变得愈发有用。在基因测序项目中，会进行大量的统计检验。在这些情况中，错误发现率可以用在检验协议中，而单个错误“发现”是指假设检验的结果（例如，在两个样本之间）。研究人员也寻求通过设置检验过程的参数去控制一定水平的错误发现率。错误发现率也适用于数据挖掘的分类场景中，其中的错误发现是指对单个记录的错误标记，特别是将 0 误标记为 1（参见 5.4.2 节）。

出于多种原因，尤其包括“多重性”这一常见问题，更多的研究并不一定意味着更好的研究。2011 年，拜耳制药公司试图对 67 项科学研究进行复现时，发现只能完全复现其中的 14 项。有近三分之二的研究根本无法复现。

在任何情况下，针对高度定义和结构化的统计检验的校正过程过于特定，也不够灵活，因此通常并不适用于数据科学家。就多重性问题而言，数据科学家的底线做法如下。

- 对于预测建模，可以通过交叉验证（参见 4.2.3 节）和使用验证集降低得到虚假模型的风险。虚假模型的效能在很大程度上是随机性的结果。
- 对于其他过程，如果没有已标记的验证集可以验证模型，那么必须依赖如下原则。
  - 应意识到对数据的查询和操作越多，随机性可能发挥的作用就更大。
  - 使用重抽样和模拟等启发式方法，为随机性提供基准测试。这样就可以将观察到的结果与基准测试进行比较。

本节要点

- 在研究工作或数据挖掘项目中，多重性（多重比较、多变量、多模型等）增加了仅根据随机对某个结果得出显著性结论的风险。
- 对于涉及多重统计比较的情况（即显著性的多重检验），可以使用统计校正过程。
- 在数据挖掘中使用结果变量带标记的验证样本，有助于避免得到误导性的结果。

拓展阅读

- David Lane 的在线统计教程中简要介绍了如何使用 Dunnett 检验校正多重比较。
- Megan Goldman 对 Bonferroni 校正做了更详细的解释，参见 http://www.stat.berkeley.edu/~mgoldman/Section0402.pdf。
- 要深入了解如何使用更灵活的统计过程调整 p 值，推荐阅读 Peter Westfall 和 Stanley Young 合著的 *Resampling-Based Multiple Testing* 一书。
- 关于数据分区和在预测建模中使用验证样本的讨论，请参阅 Galit Shmueli、Peter Bruce 和 Nitin Patel 合著的 *Data Mining for Business Analytics* 一书的第 2 章。

3.7 自由度

在许多统计检验的文档和设置中，我们都能看到**自由度**这一概念。自由度应用于从样本数据计算得到的统计量，指可以自由变化的值的个数。例如，对于一个具有 10 个值的样本，如果知道了样本的均值以及样本中的 9 个值，那么第 10 个值也是已知的，即只有 9 个值是自由变化的。

主要术语

***n*，即样本规模**

在数据中，观测（也称为**行**或**记录**）的数量。

d.f.

degrees of freedom（自由度）的简写。

自由度是很多统计检验的一个输入。例如，在计算方差和标准偏差时，分母 $n-1$ 就是自由度。为什么要使用自由度？在使用一个样本估计总体的方差时，如果在分母上使用了 n，那么估计的偏差就会偏小。如果在分母上使用了 $n-1$，这时估计就是无偏的。

t 检验、F 检验等各种对假设的标准检验，占据了传统统计学课程或教材的大部分内容。在传统的统计学公式中，如果使用了经过归一化的样本统计量，自由度就是归一化计算的一部分，它确保了归一化的数据可以匹配适当的参考分布，如 t 分布、F 分布等。

自由度对数据科学是否也同样重要？答案是并非如此，至少就显著性检验而言并非如此。一方面，在数据科学中，我们只是保守地使用了正式的统计检验。另一方面，数据的规模通常会非常大，这使得对于数据科学家来说，分母是 n 还是 $n-1$ 几乎没有区别。

但是在数据科学中，有一种场景是与自由度相关的，那就是在回归（包括逻辑回归）中使用因子化变量。如果在回归算法中使用了完全冗余的预测变量，那么算法就会产生阻塞。该问题经常出现在将分类变量因子化为二元标识（虚拟变量）的情况下。以星期为例，虽然一个星期有 7 天，但具体是星期几，其自由度为 6。一旦我们知道某一天并不是从星期一到星期六中的任意一天，那么它一定是星期天。因此，如果在回归中包括了星期一至星期六，就意味着也加入了星期天，而由于**多重共线性**（multicollinearity）问题，这将导致回归失败。

本节要点

- 自由度是归一化检验统计量计算的一部分。它使得归一化后的结果可以与参考分布（例如 t 分布、F 分布等）进行对比。
- 在回归中，为避免出现多重共线性问题，在将分类变量因子化为 n–1 个标识或虚拟变量时，应考虑其中隐含的自由度概念。

拓展阅读

几个介绍自由度的网络教程。

3.8　方差分析

如果我们不是要对两个组做 A/B 测试，而是要对比多个包含数值型数据的组（比如 A、B、C、D），这时可以使用**方差分析**（ANOVA，analysis of variance）。方差分析是一种检验多个组之间统计显著性差异的统计学方法。

主要术语

两两对比

对于有多个组的情况，在两个组之间做假设检验（比如对均值）。

多项检验（omnibus test）

一种可以测定多个组均值间方差的单一假设检验。

方差分解

从整体统计量中（例如，从整体均值、处理均值以及残差中），分离出单个值的贡献情况。

***F* 统计量**

一种归一化统计量，用于衡量多个组均值间的差异是否会超过随机模型的预期。

SS

sum of square（平方和）的简写，指与某一均值的偏差。

表 3-3 显示了 4 个 Web 页面的黏性，体现为在页面上停留的秒数。这 4 个页面是随机切换的，因此每位 Web 访问者都是随机地访问其中一个页面。每个页面总共有 5 位访问者，表 3-3 中的每一列都是一组独立的数据。第一个页面的首位访问者与第二个页面的首位访问者间并无关联。注意，在此类网络测试中，是无法完全实现经典的随机抽样设计的，即无法做到每位访问者都是从庞大的总体中随机选择的。一旦有访问者访问了一个页面，我们就记录该访问者。访问者之间可能存在一些系统性差异，具体取决于一天中的到访时间、一周中的到访日、一年中的到访季节、访问者的网络状况、访问者使用的设备等因素。在审核实验结果时，这些因素都应被视为潜在的偏差。

表3-3：4个Web页面的黏性（单位：秒）

	页面1	页面2	页面3	页面4
	164	178	175	155
	172	191	193	166
	177	182	171	164
	156	185	163	170
	195	177	176	168
平均值	172	185	176	162
总平均值				173.75

现在，我们面对着一个难题（如图 3-6 所示）。如果我们只对两个组做比较，那么事情非常简单，只需查看各组均值间的差异即可。对于 4 组均值，存在如下 6 种可能的组间比较。

- 页面 1 与页面 2 相比
- 页面 1 与页面 3 相比
- 页面 1 与页面 4 相比
- 页面 2 与页面 3 相比
- 页面 2 与页面 4 相比
- 页面 3 与页面 4 相比

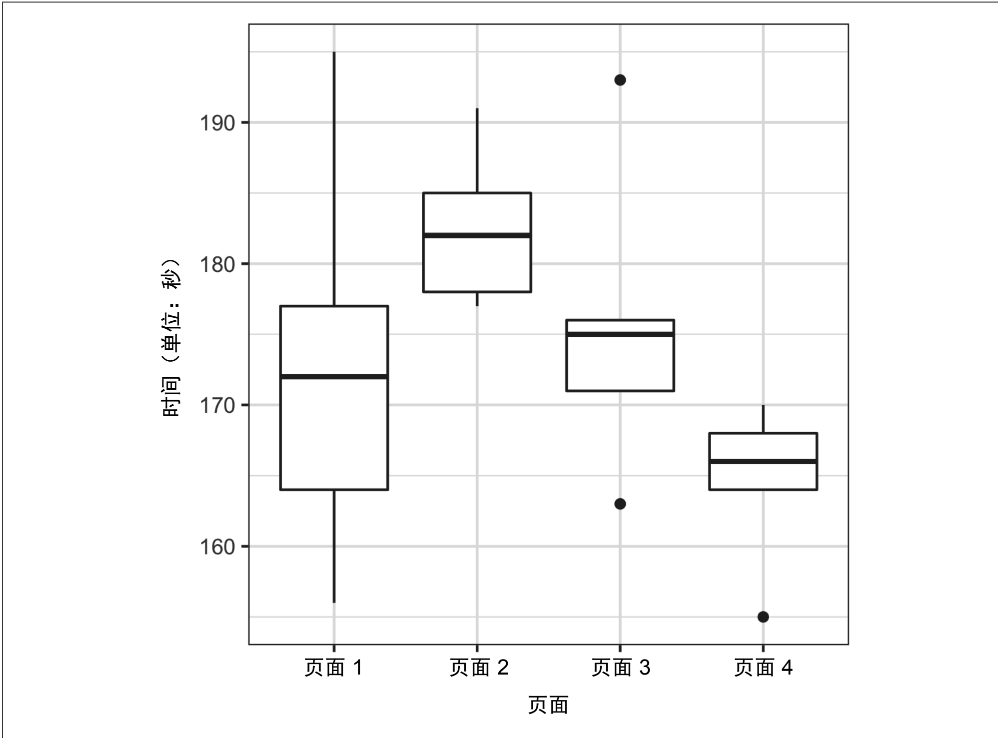


图 3-6：4 组的箱线图显示了组间的显著差异

所做的**两两**比较越多，我们就越有可能被随机性愚弄（参见 3.6 节）。我们无须比较各个页面之间所有可能的对比方式，而是可以通过整体使用单一的**多项检验**来解决这一问题："所有的页面是否具有相同的黏性？它们之间的差异是不是由于在 4 个页面间随机地分配了同一组会话时间所导致的？"

这里我们使用的检验过程就是方差分析。下面列出对 Web 页面黏性做 A/B/C/D 测试的重抽样过程，我们可以从中看到方差分析的基础所在。

(1) 将所有数据合并成一个箱子。
(2) 混洗，并从箱子中抽出 4 组样本，每组样本有 4 个值。
(3) 记录每组的均值。
(4) 记录 4 个均值间的方差。
(5) 重复第 2 步到第 4 步多次（例如 1000 次）。

这样，重抽样方差超过观测方差的比率，就是 p 值。

这里给出的置换检验比 3.3.1 节介绍的置换检验略微复杂。幸运的是，我们可以直接使用 `lmPerm` 软件包提供的 `aovp` 函数实现置换检验的计算。

```
> library(lmPerm)
> summary(aovp(Time ~ Page, data=four_sessions))
[1] "Settings: unique SS "
Component 1 :
            Df R Sum Sq R Mean Sq Iter Pr(Prob)
Page         3    831.4    277.13 3104  0.09278 .
Residuals   16   1618.4    101.15
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

输出结果中，`Pr(Prob)` 列显示的是 p 值，此例中为 0.09278。`Iter` 列显示了置换检验的迭代次数[3]。其他列分别对应于传统 ANOVA 表中的相应列，我们将在本节后面介绍。

3.8.1 *F*统计量

在比较两组的均值时，我们可以使用 t 检验替代置换检验。类似地，对于方差分析而言，存在一种基于 ***F* 统计量**的统计检验。F 统计量基于各组均值间的方差（即处理效果）与由于残差所导致的方差间的比率。比率越高，结果就越统计显著。如果数据遵循正态分布，那么根据统计学理论，统计量也应符合某种分布。由此，p 值也是可以计算的。

在 R 语言中，可以使用 `aov` 函数计算 ANOVA 表。

```
> summary(aov(Time ~ Page, data=four_sessions))
            Df Sum Sq Mean Sq F value Pr(>F)
Page         3  831.4   277.1    2.74 0.0776 .
Residuals   16 1618.4   101.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

注 3：实际运行 `avop` 函数时，给出的 `Iter` 和 `Pr(Prob)` 会与本例中的输出值存在一定的差异。——译者注

在输出结果中，`Df` 表示自由度，`Sum Sq` 表示平方和，`Mean Sq` 是"均方偏差"（mean-squared deviations 的简写），`F Value` 是 *F* 统计量。总平均的平方和等于总平均（与 0 的差值）平方后再乘以观测数 20。根据定义，总平均的自由度为 1，处理均值的自由度为 3（因为一旦设定了三个处理值，那么总平均也就设定了，这样另一个处理的均值就不会改变）。处理均值的平方和是处理均值与总平均间差值的平方和。残差的自由度为 20（即所有的观测值都可以变化），*SS* 是单个观测值与处理均值间差值的平方和。均方根（*MS*）是平方和除以自由度。*F* 统计量是处理的平方和除以误差的平方和。*F* 值仅取决于 *F* 统计量，并且参考标准的 *F* 分布，以确定处理均值间的差异是否会大于随机变异的预期差异。



方差分解

数据集的观测值可以看成多个成分的总和。对于一个数据集中的任意一个观测值，可以分解为总平均、处理效果和残差。我们将这一过程称为**方差分解**。

(1) 从总平均开始（对于 Web 页面黏性数据，总平均为 173.75）。

(2) 加入处理效果，它可能为负值（对于 Web 页面黏性数据，独立变量为 Web 页面）。

(3) 加入残差。残差也可能为负值。

这样，对 A/B/C/D 测试表（表 3-3）中左上角值（即 164）的方差分解如下。

(1) 从总平均开始：173.75。

(2) 添加处理（组）效果：组均值 172−173.75 = −1.75。

(3) 添加残差：164−172 = −8 。

(4) 得到结果：164。

3.8.2 双向方差分析

上一节介绍的 A/B/C/D 测试是一种"单向"方差分析，其中只有一个变化因子（组）。我们可以加入第二个因子，例如"周末与工作日"，并在每对组合上收集数据（周末 A 组、工作日 A 组、周末 B 组等）。这就构成一个**双向方差分析**。我们可以使用类似于单向方差分析的实现方法，通过识别"交互效应"实现双向方差分析。在确定了总平均效果和处理效果后，我们将各组中的周末和工作日观测结果分成子集，并找出各个子集的均值与处理均值之间的差异。

我们可以看到，方差分析（包括双向方差分析）是迈向完全统计模型（例如回归和逻辑回归）的第一步。完全统计模型可以对多个因子及因子的影响情况建模（参见第 4 章）。

本节要点

- 方差分析是一种用于分析多组处理结果的统计过程。
- 方差分析是对 A/B 测试中类似过程的一种扩展，用于评估各组之间的整体方差是否落在随机变异范围内。
- 方差分析的一个有用结果是识别出与组处理、交互效果和误差相关的方差成分。

3.8.3 拓展阅读

- 在 Peter Bruce 的 *Introductory Statistics and Analytics: A Resampling Perspective* 一书中，专门有一章介绍了方差分析。
- George Cobb 撰写的 *Introduction to Design and Analysis of Experiments* 一书全面介绍了方差分析，适合阅读。

3.9 卡方检验

通常在 Web 测试中，需要一次检验多个处理，这超出了 A/B 测试的范围。卡方检验适用于计数数据，它可以检验数据与预期分布的拟合程度。在统计实践中，**卡方**统计量的最常见用法是与 $r \times c$ 列联表一起使用，以评估对变量间独立性的零假设是否合理。

卡方检验最初是由卡尔·皮尔逊（Karl Pearson）在 1900 年提出的。“卡方”（Chi）一词来自皮尔逊在文章中使用的希腊字母 ξ。

主要术语

卡方统计量

观测数据偏离预期程度的量度。

期望值 / 期望

在某种假设（通常是零假设）下，我们期望数据能给出的结果。

d.f.

自由度。



$r \times c$ 表示“行数 × 列数”。例如，2×3 的表格具有两行三列。

3.9.1 卡方检验：一种重抽样方法

假设我们要对 1000 名访问者测试三种不同的标题：A、B 和 C，测试结果如表 3-4 所示。

表3-4：3种不同标题的Web检验结果

	标题A	标题B	标题C
点击	14	8	12
未点击	986	992	988

从表 3-4 中可以看到，各标题之间存在明显的差异。虽然实际的点击量很少，但是标题 A 的点击量几乎是标题 B 的两倍。重抽样过程可以检验观测到的点击量是否与随机性可导致的程度有所差异。对于 Web 标题检验，我们需要知道点击量的预期分布。在本例中，我们使用了零假设，即所有 3 种标题具有相同的点击率，这时总体点击率为 34/3000。基于该假设，我们生成了如表 3-5 所示的列联表。

表3-5：3个标题在点击率相同（零假设）情况下的期望值

	标题A	标题B	标题C
点击	11.33	11.33	11.33
未点击	988.67	988.67	988.67

我们用“Observed”表示实际观测到的情况，“Expected”表示采用假设情况下的期望值，**皮尔逊残差**（Pearson residual）R_p 的定义为：

$$R_p = \frac{\text{Observed} - \text{Expected}}{\sqrt{\text{Expected}}}$$

皮尔逊残差 R_p 测量了实际观测值与期望值之间的差异程度，如表 3-6 所示。

表3-6：皮尔逊残差

	标题A	标题B	标题C
点击	0.792	−0.990	0.198
未点击	−0.085	0.106	−0.021

卡方统计量（chi-squared statistic）是皮尔逊残差的平方和，计算公式为：

$$\xi = \sum_{i}^{r} \sum_{j}^{c} R_p^2$$

其中，r 和 c 分别是列联表的行数和列数。对于本例，卡方统计量的值为 1.666。那么它是否超出了随机模型中可能合理发生的情况呢？

我们可以使用下面给出的重抽样算法进行检验。

(1) 构造一个矩形，其中包含 34 个 1（点击数）和 2966 个 0（未点击数）。
(2) 对矩阵中数据做随机混洗，然后从中独立地抽取出三组样本，每组样本的规模为 1000，并计算每组样本中 1 的个数（点击数）。
(3) 找出各组中混洗计数和预期计数间的平方差，并将它们相加。
(4) 重复第 2 步和第 3 步多次（例如 1000 次）。
(5) 计算重抽样偏差的平方和超过观测值的频数，这就是 p 值。

使用 R 语言的 `chisq.test` 函数，就可以计算重抽样的卡方统计量。对于本例的 Web 点击数据，卡方检验计算为：

```
> chisq.test(clicks, simulate.p.value=TRUE)

        Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)

data:  clicks
X-squared = 1.6659, df = NA, p-value = 0.4853
```

上述检验表明，结果完全是由随机性获得的[4]。

注 4：正如文中所说，此处 p 值的计算具有一定的随机性，在实际运行示例程序时，得出的 p 值可能是一个与本例输出近似的值。——译者注

3.9.2　卡方检验：统计理论

统计学的渐近理论指出，卡方统计量的分布可以由**卡方分布**近似得到。适合的标准卡方分布取决于**自由度**（参见 3.7 节）。自由度与列联表的行数 r 和列数 s 有关。

$$自由度 = (r-1) \times (c-1)$$

卡方分布通常是偏斜的，右侧具有长尾。图 3-7 显示了自由度分别为 1、2、5 和 10 时的卡方分布情况。观测统计量在卡方分布中的位置越远，p 值越小。

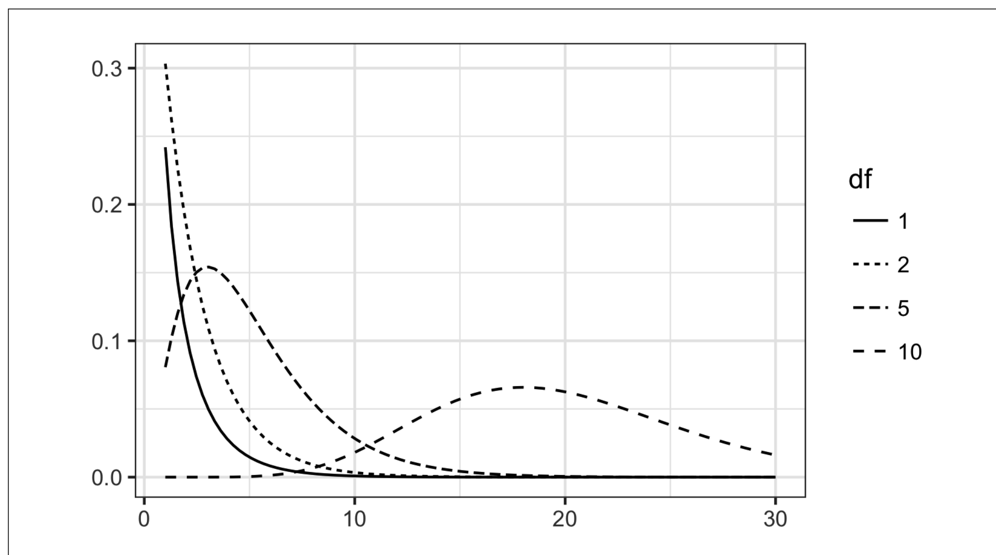


图 3-7：不同自由度下的卡方分布（y 轴为概率，x 轴为卡方统计量的值）

R 语言的 `chisq.test` 函数在计算 p 值时，使用了卡方分布作为参考分布。

```
> chisq.test(clicks, simulate.p.value=FALSE)

        Pearson's Chi-squared test

data:  clicks
X-squared = 1.6659, df = 2, p-value = 0.4348
```

此例中，卡方计算所给出的 p 值略小于重抽样的 p 值。这是因为卡方分布只是对统计量实际分布的一种近似。

3.9.3　费舍尔精确检验

卡方分布可以很好地近似上面所介绍的混洗重抽样检验过程，但是它并不适用于计数非常低（达到个位数，特别是少于 5 个）的情况。在这种情况下，重抽样过程本身就能给出更精确的 p 值。事实上，大多数统计软件都可以实际列出所有可能出现的重排（置换）情况及其频数，进而确定观测结果的极端程度。这一操作被称为**费舍尔精确检验**（Fisher's

exact test），命名源自于伟大的统计学家费舍尔（R. A. Fisher）。用 R 语言实现基本的费舍尔精确检验非常简单。

```
> fisher.test(clicks)

        Fisher's Exact Test for Count Data

data:  clicks
p-value = 0.4824
alternative hypothesis: two.sided
```

给出的 p 值为 0.4824，非常接近使用重抽样方法获得的 p 值 0.4853。

在一些情况下，虽然一些计数的值很低，但是其他的值相当高，例如转换率的分母值。由于难以计算所有可能的置换情况，这时需要做混洗置换检验，而非完全的精确检验。在上面介绍的 `fisher.test` 函数中，指定参数 `simulate.p.value=TRUE`（或 `FALSE`）就可以控制是否要使用这种近似，设置参数 `B` 的值可以控制迭代次数，而参数 `workspace` 限定了计算精确结果所使用的计算资源。

检测科学研究中的欺诈行为

一个有意义的例子来自美国塔夫茨大学的研究员 Thereza Imanishi Kari。1991 年，她被指控在研究中捏造数据，美国国会议员 John Dingell 也牵扯其中。案件最终导致她的同事 David Baltimore 辞去了洛克菲勒大学校长的职务。

虽然经过漫长的诉讼后，伊马西·卡里最终获得了豁免。但是在本案中，一个证据就来自统计学。该证据是根据实验数据中各个数字的预期分布得出的。鉴于每个观测数据都具有多个数字，调查人员关注了观测数据中各个数字的分布情况，预期是数字会遵循**统一的随机分布**。也就是说，数字是随机出现的，并且每个数字出现的概率相同（尽管首位数字可能主要取某个值，而末位数字可能会受到四舍五入的影响）。表 3-7 列出了实际数据中各个数字出现的频数。

表3-7：实验数据中各数字的出现情况

数字	频数
0	14
1	71
2	7
3	65
4	23
5	19
6	12
7	45
8	53
9	6

数据中 315 个数字的分布如图 3-8 所示。我们可以看到，这肯定不是随机出现的。

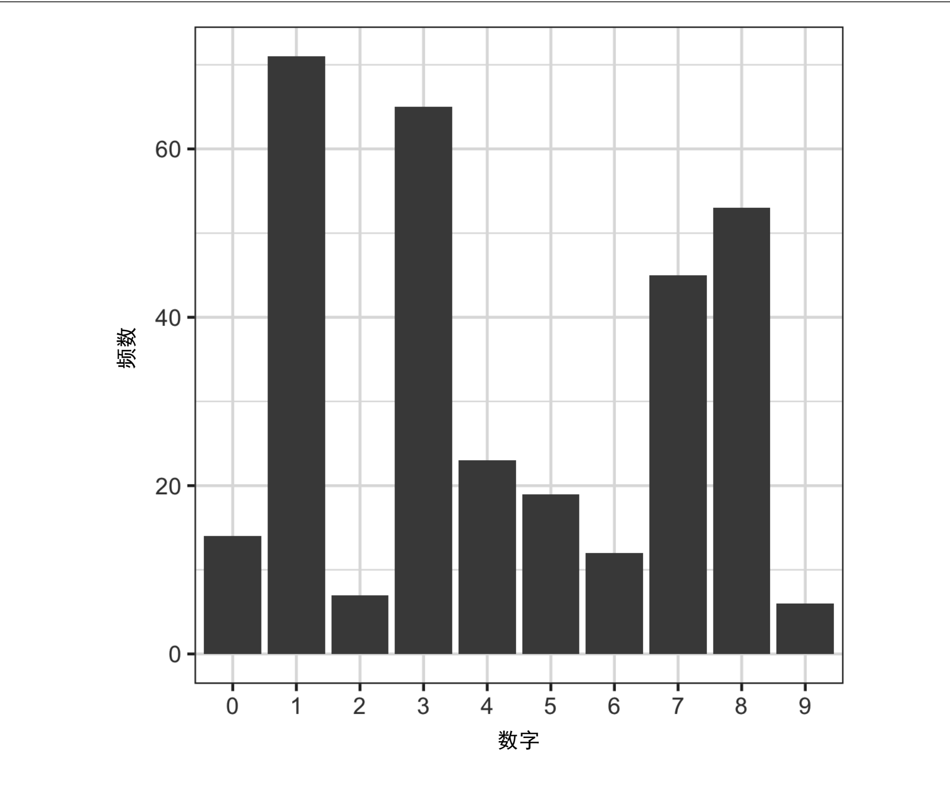


图 3-8　伊马西·卡里实验数据频数的直方图

调查人员计算了观测值与期望值的偏差情况。其中，期望值是每个数字在严格均匀分布中出现的频数，在此为 31.5。为了证明实际分布远远超出了正常随机变异的范围，调查人员使用了卡方检验（当然，也可以使用重抽样过程）。

3.9.4　与数据科学的关联

卡方检验的大多数标准用法（以及费舍尔精确检验），与数据科学的联系并不是十分紧密。在大多数数据科学实验中，无论是 A/B 测试，还是 A/B/C……测试，实验目标并不是要简单地确定统计显著性，而是要给出最佳的处理。对此，多臂老虎机算法（参见 3.10 节）可以给出更完整的解决方案。

在数据科学中，卡方检验（尤其是费舍尔精确检验）的一个应用是确定适当的 Web 实验样本规模。在此类实验中，尽管具有大量的页面展示，但是点击率通常很低。过小的计数率可能会导致实验无法得出确定的结论。这时可以使用费舍尔精确检验、卡方检验等检验方法，计算检验效能和样本规模（参见 3.11 节）。

在科学研究中，一些研究人员广泛地使用了卡方检验，以确定难以捉摸的统计显著性p值，进而使自己的研究成果适合发表。在数据科学的应用中，并不是将卡方检验或类似的重抽样模拟作为正式的显著性检验，而是更多地将此类检验方法作为一种过滤器，用以确定某个效应或特征是否值得进一步考虑。例如，此类方法可用于空间统计学和映射中，以确定空间数据是否符合某个指定的零分布，例如集中在某一区域的犯罪率是否大于随机性所允许的程度。此类方法还可以用于机器学习中的自动特征选择，通过判定各个特性的主要类是否符合随机变异的范围，即是否存在过高或过低的问题，进而确定特性的主要类。

本节要点

- 统计学中一个常见的过程是检验观测情况与独立性假设是否一致，例如购买特定产品的倾向是否与性别无关。
- 卡方分布是一种加入了独立性假设的参考分布。由观测情况计算得到的卡方统计量，必须与卡方分布进行对比。

3.9.5　拓展阅读

- 20 世纪初，费舍尔提出了“女士品茶”（Lady Tasting Tea）这一著名的例子。时至今日，该例子依然简单有效地展示了费舍尔精确检验。在网上搜索“女士品茶”，就能发现一些很好的文章。
- Stat Trek 网站提供了一个很好的卡方检验教程。

3.10　多臂老虎机算法

多臂老虎机算法（multi-arm bandit algorithm）是一种检验方法，尤其适用于 Web 测试。相比于传统的统计学实验设计方法，它实现了明显的优化，并且能更快地做出决策。

主要术语

多臂老虎机

　　一种假想的老虎机，提供多个拉杆供用户选择，每个拉杆对应不同的收益，用于模拟多处理实验。

臂

　　表示实验中的一个处理，例如 Web 测试中的标题 A。

获胜

　　通过实验模拟老虎机上的获胜，例如客户点击了链接。

传统的 A/B 测试需要根据特定的设计在实验中采集数据，去回答某个具体的问题，例如：“处理 A 和处理 B 哪个更好？”假定一旦问题得到解答，就结束实验，然后继续操作结果。

你可能已经发现，使用这一方法存在几个问题。首先，我们得到的答案并不是结论性的，即“效果未证明”。换句话说，实验结果可能会表明一个效果，但是我们没有足够的样本

去证明所表明的效果，也就无法确定效果是否符合传统的统计标准。这并未回答我们应该做出什么决策的问题。其次，我们可能希望在实验得出结论前，就开始利用之前获得的结果。再次，我们希望能根据实验结束后获得的其他数据，去更改我们的决策，或是尝试其他的事情。传统的实验方法和假设检验方法可以追溯至 20 世纪 20 年代，这些方法是相当僵化的。随着具有强大计算能力的计算机和软件的出现，我们可以使用一些更强大、更灵活的方法。此外，数据科学（包括商业）并不十分关注统计显著性，而是更加关注整体工作和结果的优化。

多臂老虎机算法在 Web 测试中广受欢迎。它可以一次测试多个处理，相比于传统的统计设计，它能更快地得出结论。该算法以赌博中使用的老虎机命名，也称“单臂老虎机”，因为该算法在配置上实现了稳定地从赌徒那里掠取金钱。让我们想象一台有多个拉杆的老虎机，每个拉杆以不同的速率付款，这就是一个多臂老虎机，即该算法全称的由来。

我们的目标是尽可能赢取更多的钱。具体地说，越早识别并确定可以获胜的拉杆越好。但是挑战在于，我们并不知道各个老虎机拉杆的回报速率，只知道拉动老虎机拉杆的结果。我们假设无论拉的是哪个拉杆，每次“获胜”将得到相同数额的回报，不同之处只在于获胜的概率。进一步假设，我们初始尝试拉动每个拉杆 50 次，得到以下结果。

- 拉杆 A：拉动 50 次，获胜 10 次。
- 拉杆 B：拉动 50 次，获胜 2 次。
- 拉杆 C：拉动 50 次，获胜 4 次。

一种极端的做法是：“拉杆 A 看起来像是赢家。因此让我们放弃尝试拉动其他的拉杆，一直拉动拉杆 B。”该做法充分利用了初始试验的结果。如果拉杆 A 的确更优，我们就可以尽早从中受益。但另一方面，如果拉杆 B 和拉杆 C 事实上更好，那么我们就会失去发现这一点的偶然性。另一种极端的做法是：“这看上去完全在随机范围内。让我们继续以均等的可能性拉动各个拉杆。”这一做法将给予拉杆 A 的替代者们一个充分展示的偶然性。但是在此过程中，我们的处理看上去并非最优。问题在于这一做法将持续多长时间？老虎机算法采用了一种混合的方法。一开始，我们更频繁地拉动拉杆 A，充分利用该拉杆初始看上去更优的结果。但我们并未放弃拉杆 B 和拉杆 C，只是较少地拉动它们。如果拉杆 A 持续表现优异，我们将继续少拉动拉杆 B 和拉杆 C，而是更频繁地拉动拉杆 A。而如果拉杆 C 的表现开始变好，拉杆 A 的结果开始变糟，这时我们可以减少拉动拉杆 A 的次数，转而频繁地拉动拉杆 C。如果其中一个拉杆被证明是优于拉杆 A 的，只是由于随机性导致它未在初始试验中显现出来，那么现在就有偶然性在进一步的检验中得以显现。

现在，我们考虑将算法应用于 Web 测试。这回测试的不再是多个老虎机拉杆，而是多个要在 Web 网站上测试的报价、标题、颜色等。用户可以点击（即商家的“获胜”），也可以不点击。初始，各个报价的展示是随机且平等的。随着测试的开展，如果一个报价开始优于其他报价，那么可以更频繁地显示该报价（即“拉动拉杆”）。但问题是，应该如何确定修改拉动速率的算法的参数？“拉动拉杆的速率”应该改成多少？何时改变速率？

下面给出了一个简单的算法，它被称为 A/B 测试的 ϵ- 贪心算法。

(1) 生成一个介于 0 和 1 之间的随机数。

(2) 如果随机数落在 0 和 ε 之间（其中，ε 是一个介于 0 和 1 之间的数字，通常非常小），则抛一次硬币（硬币是均匀的，即得到正反面的概率均为 50%）。
   a. 如果硬币正面向上，显示报价 A。
   b. 如果硬币反面向上，显示报价 B。
(3) 如果随机数大于或等于 ε，显示迄今为止具有最高响应率的报价。

ε 是控制该算法的唯一参数。如果 $\varepsilon = 1$，最终得到的是一个标准的简单 A/B 测试，每个实验对象在处理 A 和处理 B 之间随机分配。如果 $\varepsilon = 0$，最终得到一个纯粹的**贪心**算法。贪心算法无须做进一步的实验，将实验对象（Web 访问者）分配给表现最好的处理即可。

一个更复杂的算法使用了**汤普森抽样**（Thompson's sampling）方法。我们可以在每个阶段中做一次“抽样”（拉动拉杆），以最大化选择最佳拉杆的可能性。当然，我们并不知道哪个拉杆是最佳的，而问题完全在于此！但是随着每一次连续的抽取，我们都能获得收益，进而获得更多的信息。汤普森抽样采用了贝叶斯方法。它首先使用 **Beta 分布**假设回报的先验分布。Beta 分布常用于指定贝叶斯问题中的先验情况。随着每次抽取信息的累积，通过更新累积信息，我们就可以更好地优化下一次抽取，直至选取最优的拉杆。

老虎机算法可以有效地应对三种以上的处理，并趋向于最佳选择的方向。对于传统的统计检验过程而言，三种以上处理决策的复杂性，远远超出了传统的 A/B 测试，因此老虎机算法颇具优势。

本节要点

- 传统的 A/B 测试基于随机抽样过程，会导致过度地使用非最优处理。
- 相比而言，多臂老虎机算法改进了抽样过程，加入了在实验过程中学到的信息，减少了非最优处理的频数。
- 多臂老虎机算法还有助于有效地应对两种以上的处理。
- 多臂老虎机具有多种不同的算法，能够解决如何将抽样概率从非最优处理转移到（假设的）最优处理的问题。

拓展阅读

- John Myles White 撰写的 *Bandit Algorithms for Website Optimization* 一书，对多臂老虎机算法进行了很好的概要介绍。怀特在书中还提供了 Python 代码，以及评估老虎机算法性能的模拟结果。
- 关于汤普森采样的更多（技术性）介绍，参见 Shipra Agrawal 和 Navin Goyal 的论文“Analysis of Thompson Sampling for the Multi-armed Bandit Problem”。

3.11 检验效能和样本规模

在开展 Web 测试时，如何确定测试时间（即每个处理需要显示多少次）？尽管在网上可以找到很多关于如何进行 Web 测试的操作指南，但并没有一个很好的一般性指导。测试时间主要取决于实现期望目标的频数。

主要术语

效果规模

在统计检验中，期望能检测到的效果的最小规模，例如点击率提高 20%。

检验效能

给定样本规模，检测到给定效果规模的概率。

显著性水平

在检验中所使用的统计显著性水平。

在计算样本规模时，其中一个步骤是询问："一个假设检验能否真正揭示处理 A 和处理 B 之间的差异？"作为假设检验的结果，p 值不仅取决于处理 A 和处理 B 之间的真正差异，而且还取决于抽取中的运气成分，即如何选取实验组。但是，处理 A 和处理 B 之间的实际差异越大，这种差异被实验揭示的可能性也就越大；反之，如果差异越小，那么就需要更多的数据才能检测到这种差异。在棒球运动中，要区分打击率为 0.350 的击球手和打击率为 0.200 的击球手，并不需要很多的打数。而要区分打击率为 0.300 的击球手和打击率为 0.280 的击球手，则需要更多的打数。

检验效能是指在一定的样品特性（尺寸和变异性）下，检测到指定**效果规模**的概率。例如，我们可以假设在 25 个打数中，区分打击率为 0.330 的击球手和打击率为 0.200 的击球手的概率是 0.75。这时，效果规模就是 0.130（两者打击率上的差异）。而"检测"意味着假设检验会拒绝"无差异"的零假设，并得出具有实际效果的结论。因此，在两名击球手的 25 打数（$n = 25$）实验中，效果规模为 0.130，（假设）检验效能为 0.75，即 75%。

我们可以看到，检验效能中有几个步骤是可替换的，很容易加入多种所需的统计假设和公式（以指定样本的变异性、效果规模、样本规模以及用于假设检验的 α 级别等，以及计算检验效能）。事实上，已经有专门的统计软件可以计算检验效能。数据科学家在发表论文或其他工作中，大多不需要按部就班地经过各个步骤来计算检验效能。但是在某些场合下，可能还是需要收集一些 A/B 测试的数据，而数据的收集或处理会产生成本。这时，如果能大致地了解需要收集多少数据，将有助于避免出现收集了一些数据却不能得出结论性结果的情况。下面给出一种相当直观的替代做法。

(1) 从一些假设数据开始，这些数据代表了我们对所得数据的最佳猜测（可能基于先验数据）。例如，一个箱子，其中包含了 20 个 1 和 80 个 0，用于表示一名打击率为 0.200 的击球手，或者包含"在网站上花费的时间"的观测值。
(2) 在第一个样本中添加所需的效果规模，以创建第二个样本。例如，第二个箱子，其中包含了 33 个 1 和 67 个 0，或者在每个初始的"在网站上花费的时间"增加 25 秒。
(3) 从每个箱子中，抽取规模为 n 的自助样本。
(4) 对两个自助样本做置换（或基于公式的）假设检验，并记录两者之间的差异是否具有统计显著性。
(5) 重复第 3 步和第 4 步多次，并确定差异为统计显著的频数。这就是估计的检验效能。

3.11.1 样本规模

检验效能计算最常用于估计所需的样本规模。

例如，假设我们要查看点击率的情况，即点击次数占展示次数的百分比，并检验已有广告与新广告之间的对比情况。那么在此研究中，我们需要积累多少次点击？如果我们只关注能显示出巨大差异的结果（例如，50% 的差异），那么使用较小规模的样本就可以。另一方面，如果我们关注的是微小的差异情况，那么就需要规模更大的样本。一种标准方法是制定一个策略，指定新广告必须比现有广告好百分之多少（例如 10%），否则将保持现有的广告不变。这个目标就称为**效果规模**，它决定了样本的规模。

例如，假设当前的点击率约为 1.1%，而我们寻求 10% 的提升，即升至 1.21%。因此我们构建两个箱子，箱子 A 中 1 占 1.1%（例如，箱子中有 110 个 1 和 9890 个 0），箱子 B 中 1 占 1.21%（例如，箱子中有 121 个 1 和 9879 个 0）。我们先尝试从每个箱子中做 300 次抽取（对于广告而言，就是做 300 次“展示”）。假设我们第一次抽取的结果如下。

- 箱子 A：3 个 1
- 箱子 B：5 个 1

显而易见，任何假设检验都会认为这种差异（5 比 3）是在随机变异的范围之内。但是要让任意假设检验都能可靠地展示出差异情况，这里使用的样本规模（每个组中 $n = 300$）和效果规模（差异 10%）过小。

现在，我们可以尝试增大样本规模（试试展示 2000 次），并要求点击率提升更大的幅度（例如，提升 30% 而不是 10%）。

假设目前的点击率仍然是 1.1%，但我们现在它提升 50%，即提升到 1.65%。我们构建两个箱子，箱子 A 中 1 依然占 1.1%（例如，110 个 1 和 9890 个 0），而箱子 B 中 1 占 1.65%（例如，165 个 1 和 9868 个 0）。现在，我们尝试对每个箱子做 2000 次抽取。假设我们第一次抽取的结果如下。

- 箱子 A：19 个 1
- 箱子 B：34 个 1

对该差异情况（34 比 19）的显著性检验表明，尽管它比前面给出的差异（5 比 3）更接近显著，但仍然是“不显著的”。为了计算检验效能，我们需要多次重复上面的过程，或者使用可以计算检验效能的统计软件。但是我们的初始抽取表明，即便是要检测到 50% 的提升，广告也需要做上千次的展示。

总之，在计算检验效能或所需的样本规模时，有四个成分是可替换的。它们分别是：

- 样本规模
- 要检测的效果规模
- 执行检验的显著性水平，即 α 值
- 检验效能

如果指定了其中三个成分，那么就可计算得到第四个成分。最常见的情况是需要计算样本的规模，因此必须指定其他三个成分。下面的 R 代码使用 `pwr` 软件包，给出了涉及两个成

分的测试，其中两个样本的规模相同。

```
pwr.2p.test(h = ..., n = ..., sig.level = ..., power = )

h= effect size (as a proportion)
n = sample size
sig.level = the significance level (alpha) at which the test will be conducted
power = power (probability of detecting the effect size)
```

本节要点

- 在确定样本的规模之前，需提前确定要执行的统计检验。
- 必须指定要检测效果的最小规模。
- 还必须指定检测这一效果规模（检验效能）所需的概率。
- 最后，还必须指定执行检验的显著性水平（α 值）。

3.11.2 拓展阅读

- Tom Ryan 撰写的 *Sample Size Determination and Power* 一书，对此问题做出了全面的综述，适合阅读。
- 针对该问题，统计顾问 Steve Simon 以叙事风格撰写了一篇引人入胜的文章“P. Mean: The first three steps in selecting an appropriate sample size”。

3.12 小结

实验设计的原则是，将实验对象随机置入进行不同处理的两个或多个组中。良好的实验设计可以让我们对每种处理的效果得出有效的结论。在实验中，最好包括一个“不做任何改变”的对照组。虽然正式的统计推断（包括假设检验、p 值、t 检验等）占据了传统统计学课程和教材的大部分时间或空间，但是数据科学并不需要这些形式化的内容。然而，我们依然需要认识到随机变异性对人类大脑的愚弄。直观的重抽样过程（包括置换和自助法），使得数据科学家可以衡量随机变异对数据分析的影响程度。

第 4 章

回归与预测

统计学中最常见的目标可能就是回答下列问题：变量 X（很多情况下是 $X_1,\cdots,X_p$）与变量 Y 是否有关联？如果两者间有关联，那么关联的关系如何？是否可以使用这种关联关系去预测 Y？

预测是统计学与数据科学联系最为紧密的一个领域，特别是根据其他“预测”变量的值去预测结果（目标）变量。**异常检测**是两个学科紧密关联的另一个领域。尽管回归诊断最初用于数据分析和改进回归模型，但在异常检测中，回归诊断可用于检测异常的记录。对相关性和线性回归的最初使用，可追溯到一个多世纪以前。

4.1 简单线性回归

简单线性回归用于建模两个变量变化幅度间的关系。例如，Y 随着 X 的增大而增大，或者 Y 随着 X 的增大而减小[1]。相关性是衡量两个变量间相关情况的另一种方法，我们已经在 1.7 节中介绍过。这两者之间的差别在于，相关性衡量的是两个变量的关联程度，而回归则量化了两个变量间关系的本质。

主要术语

响应变量

想要预测的变量。

同义词：因变量、变量 Y、目标、结果

注 1：本章内容的版权属于本书作者彼得 · 布鲁斯和安德鲁 · 布鲁斯，© 2017 Datastats，LLC。使用需经许可。

自变量

用于预测响应的变量。

同义词：自变量、变量 X、特征、属性

记录

一个表示特定个体或实例的向量，由因子和结果值组成。

同义词：行、案例、实例、示例

截距

回归线的截距，即当 $X=0$ 时的预测值。

同义词：b_0、β_0

回归系数

回归线的斜率。

同义词：斜率、b_1、β_1、参数估计值、权重

拟合值

从回归线获得的估计值 $\hat{Y}_i$。

同义词：预测值

残差

观测值和拟合值之间的差异。

同义词：误差

最小二乘法

一种通过最小化残差的平方和而拟合回归的方法。

同义词：普通最小二乘法

4.1.1 回归方程

对于“X 发生一定的改变时，Y 的改变程度”问题，简单线性回归可以做出准确的估计。问题中的变量 X 和变量 Y 是可以互换的，只是使用的相关系数不同。对于回归问题，我们力图使用线性关系（即一条直线）从变量 X 预测变量 Y，表示为：

$$Y = b_0 + b_1 X$$

该公式表述为：“Y 等于 X 乘以 b_1，再加上常数 b_0。”其中，我们称 b_0 为**截距**”（或常量），b_1 为 X 的**斜率**。尽管“系数”这一术语通常用于 b_1，但是在 R 语言的输出中，b_0 和 b_1 都被称为**系数**。变量 Y 被称为**响应变量**或**因变量**，因为它依赖于 X。而变量 X 被称为**预测变量**或**自变量**。机器学习领域的人士习惯将 Y 称为**目标**，将 X 称为**特征向量**。

下面看一下图 4-1 中的散点图。图中显示了工人的棉尘接触年限（`Exposure`）与肺容量测

量（即呼气流速峰值，PEFR）。那么 PEFR 与 Exposure 的相关性如何？只根据图 4-1 是很难讲清楚的。

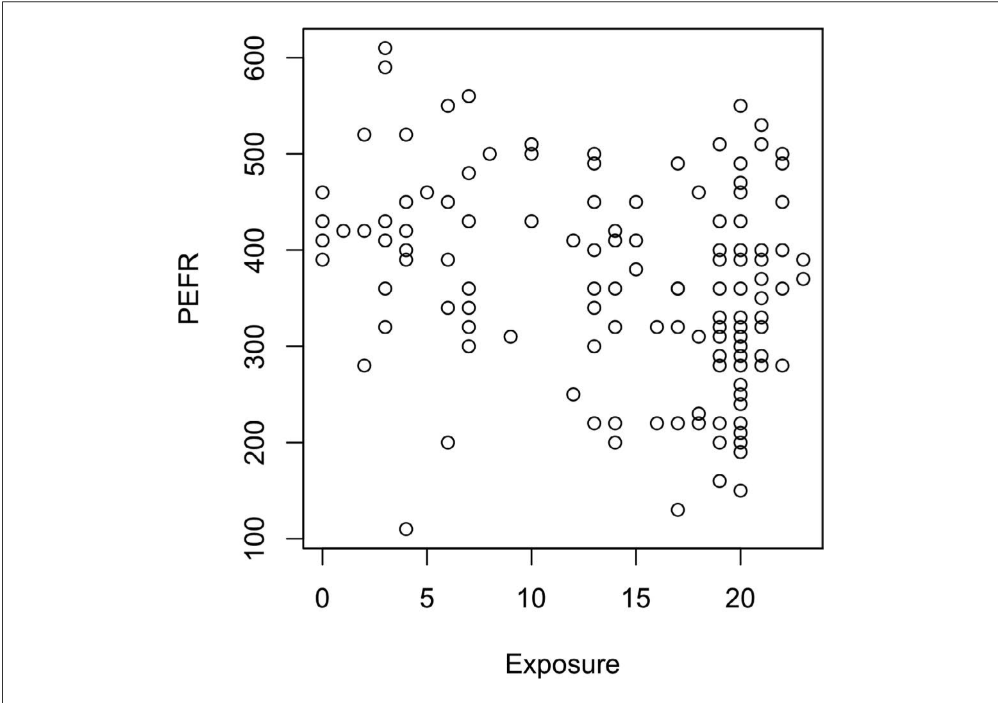


图 4-1：工人的棉尘接触年限与肺容量的散点图

简单线性回归试图找到“最优的”直线，去预测响应 PEFR 与预测变量 Exposure 之间的函数关系。

$$\text{PEFR} = b_0 + b_1 \text{ Exposure}$$

R 语言提供了 lm 函数，可用于拟合线性回归。

```
model <- lm(PEFR ~ Exposure, data=lung)
```

函数名 lm 表示**线性模型**（linear model），符号“~”表示变量 PEFR 是由变量 Exposure 预测的。

打印 model 对象，将产生如下结果。

```
Call:
lm(formula = PEFR ~ Exposure, data = lung)

Coefficients:
(Intercept)     Exposure
    424.583       -4.185
```

截距 b_0 为 424.583，可以解释为“一名未接触棉尘的工人的 PEFR 预测值”。回归系数 b_1 可以解释为“工人接触棉尘的年限每增加一年，那么他的 PEFR 测量值将降低 4.185”。

图 4-2 显示了该模型的回归线。

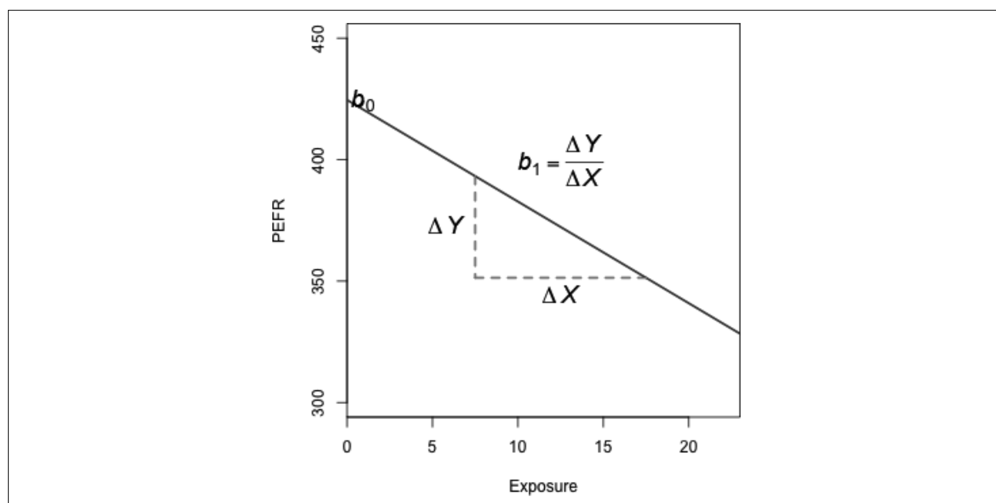


图 4-2：PEFR 数据回归拟合线的斜率和截距

4.1.2　拟合值与残差

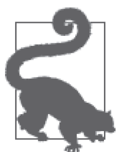
拟合值和**残差**是回归分析中的两个重要概念。一般来说，数据并不会精准地落在回归线上，因此在回归方程中，应包括一个明确的误差项 e_i：

$$Y_i = b_0 + b_1 X_i + e_i$$

我们通常使用 $\hat{Y}_i$ 表示**拟合值**，即**预测值**。拟合值的计算公式为：

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_i$$

$\hat{b}_0$ 和 $\hat{b}_1$ 表示公式中的系数不是已知的，而是估计出来的。



符号：估计值与已知值

符号用于区分估计值和已知值。因此，符号 $\hat{b}$ 表示未知参数 b 的估计值。那么，统计学家为什么要区分估计值和真实值呢？这是因为估计值具有不确定性，而真实值是固定不变的。[2]

将原始值减去**预测**值，就可以得到残差 $\hat{e}_i$。

$$\hat{e}_i = Y_i - \hat{Y}_i$$

在 R 语言中，可以使用 `predict` 函数和 `residuals` 函数计算拟合值和残差。

注 2：在贝叶斯统计学中，假定真实值为一个具有给定分布的随机变量。在贝叶斯方法中，并不是估计未知参数，而是估计先验分布和后验分布。

```
fitted <- predict(model)
resid <- residuals(model)
```

图 4-3 显示了从 PEFR 数据拟合回归线所得到的残差。残差就是图中数据和回归线间的垂直虚线的长度。

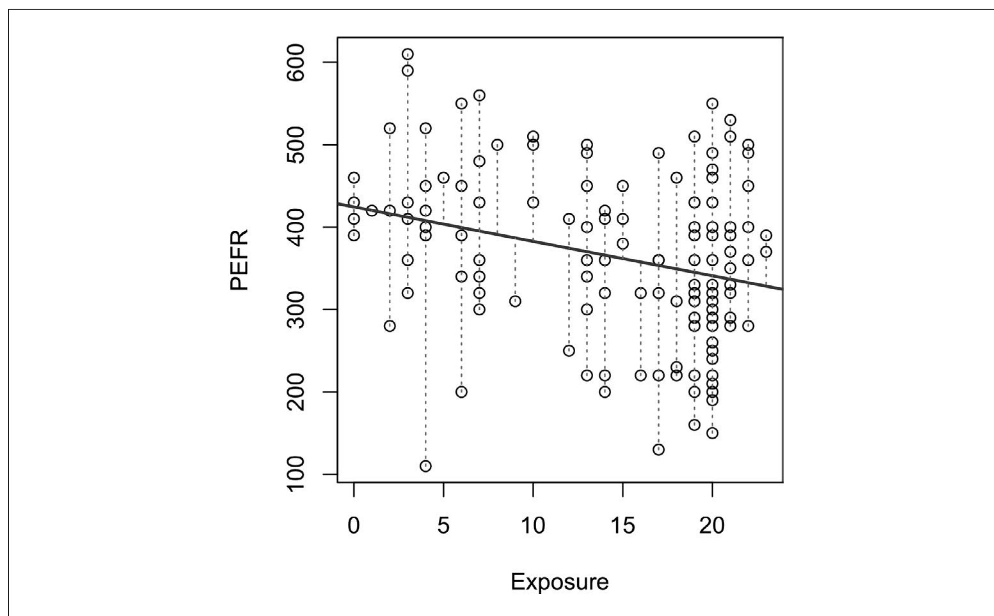


图 4-3：回归线给出的残差（注意，图 4-3 与图 4-2 的 *y* 轴尺度不同，因此斜率也明显不同）

4.1.3 最小二乘法

那么模型是如何拟合数据的？如果两者间存在清晰的关系，那么我们可以手动地拟合出一条直线。但是在实践中，回归线是使残差值的平方和最小化的估计值。残差值的平方和也称**残差平方和**（RSS），计算公式如下。

$$\begin{aligned} RSS &= \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \\ &= \sum_{i=1}^{n}(Y_i - \hat{b}_0 - \hat{b}_1 X_i)^2 \end{aligned}$$

其中，$\hat{b}_0$和$\hat{b}_1$是使 RSS 最小化的值。

我们称使得 RSS 最小化的方法为**最小二乘法**回归，或**普通最小二乘法**（OLS）回归。尽管人们一般将该方法归功于德国数学家卡尔・弗里德里希・高斯（Carl Friedrich Gauss），但它却是由法国数据家阿德里安・玛丽・勒让德（Adrien-Marie Legendre）于 1805 年最先公开发表的。最小二乘法回归给出了一种计算回归系数的简单公式。

$$\hat{b}_1 = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

$$\hat{b}_0 = \bar{Y} - \hat{b}_1\bar{X}$$

回望过去，最小二乘法之所以得到广泛的使用，一个重要原因就是该方法便于计算。随着大数据的出现，计算速度依然是一个重要因素。与均值（参见 1.3.2 节）一样，最小二乘法也对离群值敏感，但这一般只在小规模或中等规模问题中是大问题。参见 4.6.1 节对回归中的离群值的介绍。



回归术语

分析人员和研究人员在使用**回归**这一术语时，通常指的是线性回归。他们关注的是如何给出一个线性模型，去解释预测变量与数值型结果变量之间的关系。在正式的统计学意义上，回归还包括生成预测变量和结果变量之间函数关系的非线性模型。在机器学习领域中，该术语的用法偶尔也会十分宽泛，它可以指代任何生成数值型预测结果的预测模型（不同于预测二元输出或分类结果的分类方法）。

4.1.4 预测与解释（剖析）

一直以来，回归主要用于展示预测变量和结果变量之间是否存在一种假定的线性关系。回归的目标是理解变量之间的关系，并使用回归所拟合的数据去解释该关系。在这类应用中，人们关注的主要是回归方程斜率的估计值$\hat{b}$。例如，经济学家想要知道消费者支出与 GDP 增长之间的关系，公共卫生官员可能想知道公共信息运动对于提升安全性行为是否有效。这时，人们关注的并非是如何预测个别的案例，而是理解数据中的整体关系。

随着大数据的出现，回归广泛用于构建对新数据预测单个结果的模型（即预测模型），而不是解释手头已有的数据。在这类应用中，人们主要关注的是拟合值$\hat{Y}$。例如，在市场营销中，回归可用于预测收入如何随广告规模的变化而变化。一些大学也使用回归，根据学生的 STA 分数预测学生的 GPA。

尽管在一个很好地拟合了数据的回归模型中，X 的变化将导致 Y 发生变化，但是回归方程本身并未证明其中的因果关系。要得出关于因果关系的结论，必须在更宽泛的场景下理解二者之间的关系。例如，回归方程可能表明 Web 广告的点击量与会话数量之间存在一种确定性关系。但是让我们得出点击广告会提升销量这一结论的并不是回归方程，而是我们对营销过程的认知。反之则不成立。

本节要点

- 回归方程将响应变量 Y 和预测变量 X 间的关系建模为一条直线。
- 回归模型给出了拟合值和残差，即响应的预测值和预测的误差。
- 回归模型通常使用最小二乘法拟合。
- 回归可用于预测和解释。

4.1.5 拓展阅读

对预测与解释的深入介绍，可以阅读 Galit Shmueli 的论文“To Explain or to Predict”。

4.2 多元线性回归

当存在多个预测变量时，我们可以对 4.1 节中给出的回归方程做简单的扩展。

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_p X_p + e$$

现在我们得到的不再是一条直线，而是一个线性模型。在模型中，每个系数与其变量（特征）间的关系是线性的。

主要术语

均方根误差

回归均方误差的平方根，它是比较回归模型时使用最广泛的度量。

同义词：RMSE

标准残差

与均方根误差的计算一样，只是根据自由度做了调整。

同义词：RSE

***R* 方**

可以被模型解释的变异的比例，值介于 0 到 1 之间。

同义词：决定系数、R^2

***t* 统计量**

预测因子的系数，除以系数的标准误差。它提供了一种比较模型中变量重要性的度量。

加权回归

在回归中，记录具有不同的权重。

注意，简单线性回归中的所有其他概念，包括对最小二乘法拟合，以及拟合值和残差的定义，都可以扩展到多元线性回归中。例如，拟合值可以由下式给出。

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_{1,i} + \hat{b}_2 X_{2,i} + \cdots + \hat{b}_p X_{p,i}$$

4.2.1 美国金县房屋数据案例

房产估值是使用回归的一个例子。在美国，为评估某房产的税值，金县（King County）的评估师必须评估该房产的价值。通过访问 Zillow 等热门网站，购房者和专业人士可以大体了解一个合理的价格。下面使用 R 列出的数据存储在一个名为 `house` 的 `data.frame` 中。数

据是美国华盛顿金县的部分房产数据。

```
head(house[, c("AdjSalePrice", "SqFtTotLiving", "SqFtLot", "Bathrooms",
               "Bedrooms", "BldgGrade")])
Source: local data frame [6 x 6]

  AdjSalePrice SqFtTotLiving SqFtLot Bathrooms Bedrooms BldgGrade
         (dbl)         (int)   (int)     (dbl)    (int)     (int)
1       300805          2400    9373      3.00        6         7
2      1076162          3764   20156      3.75        4        10
3       761805          2060   26036      1.75        4         8
4       442065          3200    8618      3.75        5         7
5       297065          1720    8620      1.75        4         7
6       411781           930    1012      1.50        2         8
```

我们的目的是从其他多个变量中预测房屋的销售价格。lm 函数在回归方程公式右侧添加了更多的项，以实现对多元线性回归的处理。在上面的代码中，我们还要设置函数的参数 na.action=na.omit，使得模型可以丢弃那些有缺失值的记录。命令如下。

```
house_lm <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                 Bedrooms + BldgGrade,
               data=house, na.action=na.omit)
```

打印 house_lm 对象将产生如下输出。

```
house_lm

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
    Bedrooms + BldgGrade, data = house, na.action = na.omit)

Coefficients:
  (Intercept)  SqFtTotLiving        SqFtLot      Bathrooms
   -5.219e+05      2.288e+02     -6.051e-02     -1.944e+04
     Bedrooms      BldgGrade
   -4.778e+04      1.061e+05
```

对系数的解释与简单线性回归中的一样，即如果假定所有其他变量 X_k 保持不变，那么系数 b_j 就是 X_j（$k \neq j$）的单位变化所导致的预测值$\hat{Y}$的变化情况。例如，房屋的建筑面积每增加一平方英尺，房屋的估价将增加约 229 美元；如果面积增加 1000 平方英尺，那么房屋的估计值将增加 228 800 美元。

4.2.2 评估模型

从数据科学角度看，最重要的性能度量是**均方根误差**（RMSE）。均方根误差是预测值 $\hat{y}_i$ 均方误差的平方根，计算公式如下。

$$\hat{y}_i = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

均方根误差测量了模型的整体精度，是将模型与其他模型（包括使用机器学习技术拟合的

模型）对比的基础。**标准残差**（RSE）类似于均方根误差。给定 p 个预测变量，标准残差的计算公式为：

$$\text{RSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{(n - p - 1)}}$$

这两者之间的唯一差别在于，标准残差的分母是自由度，而非记录个数（参见 3.7 节）。对于线性回归而言，均方根误差和标准残差之间的差异在实践中会非常小，尤其是在大数据应用中。

可以使用 R 语言的 `summary` 函数计算一个模型的标准残差等度量。

```
summary(house_lm)

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
    Bedrooms + BldgGrade, data = house, na.action = na.omit)

Residuals:
     Min       1Q   Median       3Q      Max
-1199508  -118879   -20982    87414  9472982

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -5.219e+05  1.565e+04 -33.349  < 2e-16 ***
SqFtTotLiving  2.288e+02  3.898e+00  58.699  < 2e-16 ***
SqFtLot       -6.051e-02  6.118e-02  -0.989    0.323
Bathrooms     -1.944e+04  3.625e+03  -5.362 8.32e-08 ***
Bedrooms      -4.778e+04  2.489e+03 -19.194  < 2e-16 ***
BldgGrade      1.061e+05  2.396e+03  44.287  < 2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 261200 on 22683 degrees of freedom
Multiple R-squared:  0.5407,    Adjusted R-squared:  0.5406
F-statistic:  5340 on 5 and 22683 DF,  p-value: < 2.2e-16
```

可以看到，程序输出中有另一个有用的度量，就是**决定系数**，也称 ***R* 方**统计量，即 R^2。决定系数的取值范围在 0 ～ 1 之间，它测量了数据中可以由模型解释的变异性的比例。决定系数主要用于解释回归，它可以评估模型拟合数据的程度。决定系数的计算公式为：

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

公式中，分母值与 y 的方差成正比。在 `lm` 函数的输出中，还给出了一个**调整后的 *R* 方**值。该度量根据自由度做了调整。在多元线性回归中，它与 R 方之间几乎不存在明显的差异。

在函数的输出中，与估计系数一并给出了系数的标准误差（SE）和 t 统计量。t 统计量的计算公式为：

$$t_b = \frac{\hat{b}}{\mathrm{SE}(\hat{b})}$$

t 统计量及其镜像（即 p 值）测定了系数“统计显著”的程度，即超出预测变量和目标变量的随机分配可能生成的范围。t 统计量越大，即 p 值越低，那么预测变量的显著性越高。鉴于“简约性”（parsimony）是建模中的一个基本理念，此类工具对于指导如何选择添加到预测因子中的变量十分有用（参见 4.2.4 节）。



除了 t 统计量之外，R 和其他统计软件包通常还会给出 p 值和 F 统计量。例如，p 值在 R 语言输出中显示为 `Pr(>|t|)` 列。数据科学家一般并不关注这些统计量的解释，也不关注统计显著性的问题。数据科学家主要关注的是 t 统计量，并且使用它来指导是否需要将一个预测因子加入到模型中。如果 t 统计量很大，即 p 值接近于 0，就表示应该将预测因子保留在模型中。如果 t 统计量非常小，则表示该预测因子应该被丢弃。更多细节，参见 3.4.1 节。

4.2.3　交叉验证

经典的统计回归度量（R^2、F 统计量和 p 值）都是“样本内”（in-sample）度量，即应用于拟合模型所使用的同一数据上。我们可以直观地感受到，从原始数据中取出一些数据，并不在拟合模型时使用这些数据，这种做法是十分有意义的。随后，我们可以使用这些留出的数据（即验证集）去验证模型的效果。通常，我们可以使用大部分数据去拟合模型，然后使用余下的较小一部分数据去验证模型。

这种“样本外”（out-of-sample）验证的理念并不新颖，但是直到更大规模的数据集越来越普遍时，该理念才真正地得以实施。在使用小数据集时，分析人员总希望使用所有的数据去拟合最优模型。

然而，在使用验证样本时，我们会受限于一些不确定性，这些不确定性来自小规模验证样本的变异性。如果我们选择了不同的验证样本，那么在评估中会产生何种程度的差异呢？

交叉验证将验证样本这一理念扩展到多个依次进行验证的样本上。基本的 **k 折**（fold）**交叉验证**的算法如下。

(1) 取出 $1/k$ 的数据，作为验证样本。
(2) 用余下的数据训练模型。
(3) 将训练模型应用于验证集上（进行打分），并记录所需的模型评估指标。
(4) 将最初取出的 $1/k$ 数据放回，再取出 $1/k$ 数据，其中不包括上一次取出的任何记录。
(5) 重复第 2 步和第 3 步。
(6) 重复上述步骤，直至验证集使用了每个记录。
(7) 对模型评估度量取平均或进行组合。

上面将数据划分为训练样本和验证样本的过程，也被称为**折**。

4.2.4 模型选择和逐步回归法

在一些问题中，有很多变量可以作为回归中的预测因子。例如，要预测一处房屋的价值，可以使用房屋面积或建造年份等变量。在 R 中，很容易将这些变量添加到回归方程中。

```
house_full <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                 Bedrooms + BldgGrade + PropertyType + NbrLivingUnits +
                 SqFtFinBasement + YrBuilt + YrRenovated +
                 NewConstruction,
               data=house, na.action=na.omit)
```

但是，添加更多的变量并不意味着就会得到一个更好的模型。在模型的选择中，统计学家使用了**奥卡姆剃刀原则**（principle of Occam's razor）：在其他条件相同的情况下，应优先选用更简单的模型而不是更复杂的模型。

添加额外的变量，几乎总会降低均方根误差并增大 R^2。因此，这些统计量并不适用于指导模型的选择。20 世纪 70 年代，著名的日本统计学家赤池弘次（Hirotugu Akaike）提出了一种名为 AIC（赤池信息量准则）的指标，对给模型添加项进行了惩罚。用于回归的 AIC 的计算公式如下：

$$\text{AIC} = 2P + n \log(\text{RSS}/n)$$

其中，P 是变量的数量，n 是记录的数量。目标是找出使 AIC 最小的模型。如果模型具有 k 个额外变量，那么惩罚项为 $2k$。



AIC、BIC 和 Mallows Cp

AIC 的计算公式可能看上去颇为神秘。事实上，它基于信息理论中的渐近结果。AIC 有多个变体。

- AICc：针对小规模样本修正的 AIC。
- BIC（贝叶斯信息准则）：类似于 AIC，但是在模型中额外添加了变量，因此具有更强的惩罚。
- Mallows Cp：AIC 的一种变体，由 Colin Mallows 提出。

数据科学家通常既不需要关心上述样本内度量间的差异，也不需要关心这些度量的底层理论。

那么如何找到能使 AIC 最小的模型？一种方法是使用**全子集回归法**（all subset regression），它能搜索所有可能的模型。该方法的计算成本很高，对于具有大规模数据和大量变量的问题是不可行的。另一种替代方法更具吸引力，它使用了**逐步回归法**，通过连续地添加并丢弃预测因子，发现可降低 AIC 的模型。在由 Venebles 和 Ripley 开发的 `MASS` 软件包中提供了一个名为 `stepAIC` 的逐步回归计算函数。

```
library(MASS)
step <- stepAIC(house_full, direction="both")
step

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + Bathrooms + Bedrooms +
```

```
    BldgGrade + PropertyType + SqFtFinBasement + YrBuilt, data = house0,
    na.action = na.omit)

Coefficients:
              (Intercept)            SqFtTotLiving
               6227632.22                   186.50
                Bathrooms                 Bedrooms
                 44721.72                -49807.18
                BldgGrade  PropertyTypeSingle Family
                139179.23                 23328.69
    PropertyTypeTownhouse          SqFtFinBasement
                 92216.25                     9.04
                  YrBuilt
                 -3592.47
```

该函数选取了一个模型，其中的多个变量抽取自 `house_full:SqFtLot`、`NbrLivingUnits`、`YrRenovated` 和 `NewConstruction`。

更为简单的做法是**前向选择**（forward selection）和**后向选择**（backward selection）。在前向选择中，开始时没有预测因子，而是依次添加的。在每一步添加对 R^2 具有最大贡献的预测因子。当贡献不再统计显著时，停止继续添加。在后向选择（或**后向删除**）中，一开始就给出了一个完整的模型，然后从中逐步移除不再统计显著的预测因子，直到模型中所有预测因子都是统计显著的。

惩罚回归的思想类似于 AIC。拟合模型的函数并不是显式地搜索一组离散的模型，而是添加了一个新限制，对有多个变量（参数）的模型进行惩罚。惩罚回归不像逐步回归、前向和后向选择那样要完全清除预测变量，而是通过减少系数来应用惩罚，在一些情况下，甚至会减少至接近于 0。常见的惩罚回归是**岭回归**和 **LASSO 回归**。

对于模型的评估和调优而言，全子集回归和逐步回归是“样本内”方法。这意味着模型选取可能会受限于过拟合，不能很好地应用于新数据。为了避免出现这一问题，一种常用的方法是使用交叉验证去验证模型。在线性回归中，过拟合通常不是大问题，因为线性回归对数据给出的是一种简单（线性）全局结构。对于更为复杂的模型而言，尤其是响应本地数据结构的迭代过程，交叉验证是一种非常重要的工具，更多详细内容参见 4.2.3 节。

4.2.5　加权回归

在很多情况下，尤其是分析复杂的调查时，统计学家会使用加权回归方法。而数据科学家可能认为加权回归在下面两种情况中十分有用。

- 反方差权重（当不同观测值使用了不同的精度测量时）。
- 分析聚合的数据，加权变量编码了聚合数据中每行代表了多少个原始观测值。

以房屋数据为例，历史销售数据没有近期销售数据可靠。在使用 `DocumentDate` 确定销售年份后，我们可以计算自 2005 年（数据的开始年份）以来的年份数，以此作为 `Weight` 变量。

```
library(lubridate)
house$Year = year(house$DocumentDate)
house$Weight = house$Year - 2005
```

下面，我们使用设置了 `weight` 参数的 `lm` 函数计算加权回归。

```
house_wt <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                 Bedrooms + BldgGrade,
               data=house, weight=Weight)
round(cbind(house_lm=house_lm$coefficients,
            house_wt=house_wt$coefficients), digits=3)

                  house_lm    house_wt
(Intercept)   -521924.722 -584265.244
SqFtTotLiving     228.832     245.017
SqFtLot            -0.061      -0.292
Bathrooms      -19438.099  -26079.171
Bedrooms       -47781.153  -53625.404
BldgGrade      106117.210  115259.026
```

加权回归的系数与原始的回归系数略有差异。

本节要点

- 多元线性回归建模了响应变量 Y 与多个预测变量 $X_1, \cdots, X_p$ 之间的关系。
- 均方根误差（RMSE）和 R^2 是评价模型最重要的度量。
- 回归系数的标准误差可用于度量变量对模型的贡献的可靠性。
- 逐步回归是一种自动确定模型中应包括哪些变量的方法。
- 加权回归用于拟合函数中，可以对特定记录给予更大或更小的权重。

4.3 使用回归做预测

在数据科学中，回归的主要目的是预测。请记住这一点，因为作为一种旧有的统计学方法，回归主要用于传统的解释性建模而不是预测。

主要术语

预测区间

个体预测值的不确定区间范围。

外推法

将模型扩展到拟合所用的数据范围之外。

4.3.1 外推法的风险

回归模型不应外推到所使用的数据范围之外。回归模型仅对具有足够数据值的预测因子有效（即便是在有充足数据可用的情况下，也可能存在其他一些问题，参见 4.6 节）。举一个极端的例子，假定我们使用 `model_lm` 预测一块面积为 5000 平方英尺的空地的售价。在这种情况下，所有与建筑物有关的预测因子的值都为 0，进而回归方程会给出一个荒诞的预测值，即 $-521900 + 5000 \times (-0.0605) = -522202.5$ 美元。为什么会这样？数据中仅包含具

有建筑物的地块，并没有空地的相关记录。因此，模型不知该如何预测空地的售价。

4.3.2 置信区间和预测区间

许多统计量涉及对变异性（不确定性）的测量和理解。回归输出中的 t 统计量和 p 值以一种正式的方式处理该问题，有时这对于变量的选择十分有用（参见 4.2.2 节）。另一种更有用的指标就是置信区间，它是围绕回归系数和预测的不确定性区间。要理解置信区间，一种简单的方式是使用自助法（使用自助过程的详细信息，参见 2.4 节）。在各种统计软件的输出中，最常见的回归置信区间是回归参数（系数）的置信区间。下面给出的自助算法（bootstrap algorithm），可以对具有 P 个预测因子和 n 条记录（行）的数据集，生成一个回归参数（系数）的置信区间。

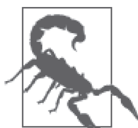
(1) 将每行数据（包括结果变量）作为一张“票”（ticket），并将所有的 n 张票置于同一个箱子中。
(2) 从箱子中随机抽取一张票，记录票上的值，并将票放回箱子。
(3) 重复第 2 步 n 次，得到一个自助法重抽样。
(4) 对自助样本做回归拟合，记录估计的回归系数。
(5) 重复第 2 步到第 4 步多次（例如 1000 次）。
(6) 现在，每个回归系数有 1000 个自助值。找到每个系数的百分位数。例如，对于 90% 置信区间是第 5 百分位数和第 95 百分位数。

要生成回归系数的实际自助置信区间，可以使用 R 的 `Boot` 函数，也可以简单地使用基于公式的置信区间，这是 R 的惯常输出。两者在概念上的意义和解释是一样的，它们对于数据科学家来说并不十分重要，他们关心的是回归系数。数据科学家更感兴趣的是围绕预测变量 y 值（即 $\hat{Y}_i$ ）的置信区间。围绕 $\hat{Y}_i$ 的不确定性来自两个方面：

- 相关的预测因子及其回归系数的不确定性（参见本节前面的自助算法）
- 单个数据点固有的额外误差

单个数据点的误差可以理解为：即便我们确切地知道一个回归方程（例如，如果我们有大量的记录可以拟合出回归方程），对于一组给定的预测因子值，回归方程的实际结果值也会存在一些变化。例如，有 8 个房间、3 间浴室和 1 个地下室，且面积为 6500 平方英尺的几个房屋，在房价上可能也会存在一些差异。我们可以使用拟合值的残差去建模单个数据点的误差。对回归模型误差和单个数据点误差建模的自助算法如下。

(1) 从数据中抽取出一个自助样本（本书已经给出了详细的做法）。
(2) 拟合回归，并预测新的值。
(3) 从原始回归拟合中随机取出一个残差，添加到预测值中，并记录结果。
(4) 重复第 1 步到第 3 步多次（例如，1000 次）。
(5) 找出结果的第 2.5 百分位数和第 97.5 百分位数。



预测区间还是置信区间？

预测区间涉及围绕单个值的不确定度，而置信区间则与由多个值计算得到的统计量（如均值）相关。因此，对于同一个值，预测区间的范围通常要比置信区间宽一些。为了在自助模型中建模单个值的误差，需要选择单个残差去处理预测值。这时，我们应该选用两者中的哪一个？这取决于具体的分析场景和目的。但是，数据科学家通常关注的是特定的单个预测，因此预测区间更适用。如果在应该使用预测区间时使用了置信区间，将严重低估给定预测值的不确定度。

本节要点

- 超出数据范围的外推会导致误差。
- 置信区间量化了回归系数的不确定度。
- 预测区间量化了单个预测中的不确定度。
- 包括 R 在内的很多统计软件，都会使用公式在默认或指定输出中给出预测区间和置信区间。
- 也可以使用自助法确定置信区间，该做法的解释和理念同上。

4.4　回归中的因子变量

因子变量（factor variable）也称为**分类变量**，它是一组数量有限的离散值。例如，贷款目的可以是“债务合并”“办婚礼”“购买汽车”等。因子变量的一种特殊情况是二元（即是 / 否）变量，也称为**指示变量**。回归需要数值输入，因此，要在回归模型中使用因子变量，需要对因子变量进行重新编码。最常用的编码方法是将因子变量转换为一组二元**虚拟**变量。

主要术语

虚拟变量

二元的 0/1 变量，通过对因子数据重新编码得到，可用于回归模型或其他模型。

参考编码

统计学家最常使用的编码类型。它以因子的一层作为参考层，并将其他因子与参考层进行对比。

同义词：编码处理

独热编码（one hot encoder）

机器学习领域中常用的一种编码。它保留了所有的因子层。虽然该编码适用于部分机器学习算法，但并不适用于多元线性回归。

偏差编码

在编码中用于对比的并不是参考层，而是将每一层与整体均值进行对比。

同义词：总和对照（sum contrasts）编码[3]

4.4.1 虚拟变量的表示

在美国金县房屋数据中，有一个因子变量表示房屋的所有权类型。下面列出了数据集中一个由六条记录组成的小子集。

```
head(house[, 'PropertyType'])
Source: local data frame [6 x 1]

   PropertyType
         (fctr)
1     Multiplex
2 Single Family
3 Single Family
4 Single Family
5 Single Family
6     Townhouse
```

在该例中，因子变量的可能取值（即“层”）有 3 个，即 `Multiplex`、`Single Family` 和 `Townhouse`。如果要使用该因子变量，需要将其转换为一个二元变量集合。我们的做法是将因子变量的每个可能取值转换为一个二元变量。这可以使用 R 提供的 `model.matrix` 函数实现[4]。

```
prop_type_dummies <- model.matrix(~PropertyType -1, data=house)
head(prop_type_dummies)
  PropertyTypeMultiplex PropertyTypeSingle Family PropertyTypeTownhouse
1                     1                         0                     0
2                     0                         1                     0
3                     0                         1                     0
4                     0                         1                     0
5                     0                         1                     0
6                     0                         0                     1
```

函数 `model.matrix` 将 R 的 `data.frame` 对象转换为一个适用于线性模型的矩阵对象。因子变量 `PropertyType` 具有三个不同的值，因此表示为一个具有三列的矩阵。这种表示在机器学习领域被称为**独热编码**（参见 6.1.3 节）。在一些机器学习算法中，例如近邻算法和树模型中，独热编码是因子变量的标准表示方式（参见 6.2 节）。

在回归中，一个具有 P 个层的因子变量，通常会使用一个只有 P−1 列的矩阵表示。这是因

注 3：该编码是一种对照编码。采用这种编码的回归方程，其回归系数之和为 0，因此得名“sum contrasts”。——译者注

注 4：`model.matrix` 函数中的 `-1` 参数生成了独热编码表示。因为要移除截距，所以是“−”。否则，R 默认会生成一个具有 P−1 列的矩阵，其中使用首个因子层作为参考层。

为回归模型中通常包括一个截距项。因此，一旦已经定义了 $P-1$ 个二元值，那么由于截距项的存在，第 P 个值就是已知的，可以看成是冗余的。如果添加了第 P 个列，将导致多重共线性错误（参见 4.5.2 节）。

R 默认使用首个因子层作为**参考**，并相对于该层去解释其余的层。

```
lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
+      Bedrooms +  BldgGrade + PropertyType, data=house)

Call:
lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
    Bedrooms + BldgGrade + PropertyType, data = house)

Coefficients:

              (Intercept)              SqFtTotLiving
               -4.469e+05                  2.234e+02
                  SqFtLot                  Bathrooms
               -7.041e-02                 -1.597e+04
                 Bedrooms                  BldgGrade
               -5.090e+04                  1.094e+05
PropertyTypeSingle Family      PropertyTypeTownhouse
               -8.469e+04                 -1.151e+05
```

R 回归的输出给出了两个回归系数，分别对应于 `PropertyTypeSingleFamily` 和 `PropertyType-Townhouse`。输出中并没有对应于 `Mutliplex` 层的回归系数。这是因为当 `PropertyTypeSingle-Family == 0` 和 `PropertyTypeTownhouse == 0` 时，已经隐式地定义了 `Mutliplex` 层。回归系数的解释是相对于 `Mutliplex` 这一因子层的。因此，`Single Family` 房屋的价值低了将近 85 000 美元，而 `Townhouse` 房屋的价值低了 150 000 多美元[5]。



各种因子编码

存在多种不同的因子变量编码方法，它们统称为**对照编码**（contrasting coding）系统。例如，**偏差编码**就是一种对照编码方式，也称为**总和对照编码**，它将每一层与整体均值做对比。**多项式编码**（polynomial coding）是另一种对照编码方式，适用于有序因子，参见 4.4.3 节。除非是对于有序因子，否则数据科学家一般只会用到参考编码或独热编码。

4.4.2　多层因子变量

一些因子变量会生成大量的二元虚拟编码——邮政编码是一个因子变量，而美国有 4.3 万个邮政编码。在这种情况下，我们可以通过探索数据了解预测变量与结果之间的关系，进而确定分类中是否包含了有用的信息。如果包含，那么我们必须进一步决定保留所有的因子是否有用，或者是否应该合并一些因子层。

例如，在金县，82 个邮政编码区域有房屋销售数据。

注 5：这种做法并不直观，但是我们可以使用 `location` 变量作为混淆变量，通过该变量的影响进行解释。参见 4.5.3 节。

```
table(house$ZipCode)

 9800 89118 98001 98002 98003 98004 98005 98006 98007 98008 98010 98011
    1     1   358   180   241   293   133   460   112   291    56   163
98014 98019 98022 98023 98024 98027 98028 98029 98030 98031 98032 98033
   85   242   188   455    31   366   252   475   263   308   121   517
98034 98038 98039 98040 98042 98043 98045 98047 98050 98051 98052 98053
  575   788    47   244   641     1   222    48     7    32   614   499
98055 98056 98057 98058 98059 98065 98068 98070 98072 98074 98075 98077
  332   402     4   420   513   430     1    89   245   502   388   204
98092 98102 98103 98105 98106 98107 98108 98109 98112 98113 98115 98116
  289   106   671   313   361   296   155   149   357     1   620   364
98117 98118 98119 98122 98125 98126 98133 98136 98144 98146 98148 98155
  619   492   260   380   409   473   465   310   332   287    40   358
98166 98168 98177 98178 98188 98198 98199 98224 98288 98354
  193   332   216   266   101   225   393     3     4     9
```

ZipCode 是一个重要的变量，它代表了地段对房价的影响。如果要包括所有的层，那么需要 81 个回归系数，对应于 81 个自由度。而原始模型 house_lm 只有 5 个自由度，参见 4.2.2 节。而且我们发现在部分邮政编码区域中，只有一个房屋销售数据。在一些问题中，鉴于邮政编码的前两位或前三位对应于次级地理区域，我们可以使用前几位数字整合邮政编码区域。但是对于金县而言，几乎所有的销售都发生在邮政编码“980*XX*”或“981*XX*”的区域中，所以这种方法并不适用。

另一种方法是根据其他变量（例如销售价格）的情况对邮政编码进行分组。更好的做法是使用初始模型的残差来构建邮政编码组。下面的 dplyr 代码基于 house_lm 回归残差的中位数，将 82 个邮政编码整合为 5 个组。

```
zip_groups <- house %>%
  mutate(resid = residuals(house_lm)) %>%
  group_by(ZipCode) %>%
  summarize(med_resid = median(resid),
            cnt = n()) %>%
  arrange(med_resid) %>%
  mutate(cum_cnt = cumsum(cnt),
         ZipGroup = ntile(cum_cnt, 5))
house <- house %>%
  left_join(select(zip_groups, ZipCode, ZipGroup), by='ZipCode')
```

代码计算每个邮政编码的中位数残差，并使用 ntile 函数按中位数排序，将邮政编码划分为 5 个组。4.5.3 节中的例子展示了如何在回归中使用这样转换的因子变量作为数据项，实现对初始拟合情况的改进。

使用残差帮助指导回归拟合的理念，是建模过程中的一个基础步骤，参见 4.6 节。

4.4.3 有序因子变量

一些因子变量体现出了因子的层级，被称为**有序因子变量**或**有序分类变量**。例如，贷款等级包括 A、B、C 等，每一级别的风险都要比前一级别高。有序因子变量通常可以转换为数值，并当作数值使用。例如，变量 BldgGrade 就是一个有序因子变量。表 4-1 给出了该

变量所代表的部分等级类型。鉴于每个等级都具有特定的意义，因此数值是从低到高排序的，对应于房屋等级的逐步提高。如果使用 4.2 节中拟合的回归模型 `house_lm`，我们需要将 `BldgGrade` 作为数值型变量处理。

表4-1：一个典型的数据格式

数值	描述
1	Cabin
2	Substandard
5	Fair
10	Very good
12	Luxury
13	Mansion

将有序因子作为数值变量处理，可以保留次序关系中所包含的信息。否则，这些信息将在转换为因子的过程中丢失。

本节要点

- 因子变量需要转换为数值变量，才能在回归中使用。
- 要编码一个具有 P 个不同值的因子变量，最常用的方法是表示为 $P-1$ 个虚拟变量。
- 即便是在规模非常大的数据集中，多层因子变量也需整合为具有更少层的变量。
- 一些因子的层是有序的，可以表示为单一的数值变量。

4.5 解释回归方程

在数据科学中，回归最重要的用途就是预测因变量（结果变量）。但是在某些情况下，从回归方程本身获得一些洞见，以理解预测变量与结果之间的关系是十分有价值的。本节将为如何查看并解释回归方程提供一些指导。

主要术语

相关变量

当预测变量高度相关时，难以解释单个回归系数。

多重共线性

当预测变量间存在完美的或近乎完美的相关性时，回归是不稳定的，或者说是不可能计算的。

同义词：共线性

混淆变量

一种重要的预测变量。忽视该变量可导致回归方程给出伪关系。

主效应

预测变量和结果变量之间的关系，该关系独立于其他的变量。

交互作用

两个或两个以上预测变量和响应之间的相互依赖关系。

4.5.1　相关的预测变量

在多元回归中，预测变量通常是相互关联的。例如，下面查看一下 4.2.4 节中拟合的 `step_lm` 模型的回归系数。

```
step_lm$coefficients
              (Intercept)             SqFtTotLiving
             6.227632e+06              1.865012e+02
                Bathrooms                  Bedrooms
             4.472172e+04             -4.980718e+04
                BldgGrade PropertyTypeSingle Family
             1.391792e+05              2.332869e+04
    PropertyTypeTownhouse            SqFtFinBasement
             9.221625e+04              9.039911e+00
                  YrBuilt
            -3.592468e+03
```

我们看到，`Bedrooms` 的回归系数竟然是负值。这意味着在房子中增加一间卧室，反而会降低房屋的价值。为什么会发生这种情况？这是因为预测变量是相互关联的。面积大的房子一般有更多的卧室，而房屋的价值受面积大小的影响，而非卧室的数量。对于两个面积相同的房子，我们更喜欢的通常不是卧室更多但面积更小的那个。

如果预测变量是相互关联的，那么回归系数的符号和值会难以解释（并且会提高估计量的标准误差）。卧室、房屋面积和卫生间数量等变量就是相关的。下面的例子展示了这一关联。我们在回归方程中移除了 `SqFtTotLiving`、`SqFtFinBasement` 和 `Bathrooms` 变量，拟合了另一个回归。

```
update(step_lm, . ~ . -SqFtTotLiving - SqFtFinBasement - Bathrooms)

Call:
lm(formula = AdjSalePrice ~ Bedrooms + BldgGrade + PropertyType +
    YrBuilt, data = house0, na.action = na.omit)

Coefficients:
              (Intercept)                   Bedrooms
                  4834680                      27657
                BldgGrade  PropertyTypeSingle Family
                   245709                     -17604
    PropertyTypeTownhouse                    YrBuilt
                   -47477                      -3161
```

在上面的代码中，`update` 函数用于为模型添加或移除变量。从输出中可以看到，现在卧室的回归系数为正了，这符合我们的预期（但既然这些变量已被移除，它实际上是房屋面积

的代理变量）。

相关变量只是回归系数解释中可能碰到的问题之一。在 `house_lm` 模型中，并没有变量表示房屋的地段信息，而且模型将不同类型的地段混淆在一起。这样，地段变量可能会成为一个**混淆变量**。关于混淆变量的更多内容，参见 4.5.3 节。

4.5.2　多重共线性

相关变量的一种极端情况就是在预测变量间存在冗余，这被称为**多重共线性**问题。如果一个预测变量可以表示为其他变量的一种线性组合，就产生了完美的多重共线性问题。产生多重共线性的情况有下面几种。

- 在误差中多次包含同一个变量。
- 从一个因子变量创建了 P 个虚拟变量，而非 $P-1$ 个虚拟变量（参见 4.4 节）。
- 两个变量近乎完美相关。

回归中的多重共线性问题必须解决掉。具体做法是依次移除变量，直至去除了多重共线性问题。如果存在完成的多重共线性，那么表明回归并没有定义良好的解决方案。包括 R 在内的很多软件包，都会自动处理某些类型的多重共线性。例如，在金县房屋数据 `house` 的回归中，两次包括了 `SqFtTotLiving` 变量，得到的结果与 `house_lm` 模型给出的一样。对于非完美的多重共线性，统计软件也许能提供一个解决方案，但这样的结果可能不稳定。



对于树模型、聚类和最近邻等非回归方法，多重共线性可能并不会构成问题。在这些非回归方法中，可能会建议保留 P 个虚拟变量，而非 $P-1$ 个。这就是说，即便是在这些方法中，非冗余的预测变量可能依然是个优点。

4.5.3　混淆变量

对于相关变量，问题在于“委任”，即回归方程中包括了多个与响应变量具有相似预测关系的变量。而对于**混淆变量**，问题在于“遗漏”，即回归方程中未能包括某个重要的变量。对回归方程相关系数的朴素解释，可能会得出一个无效的结论。

以 4.2.1 节中的金县回归方程 `house_lm` 为例。在该回归方程中，`SqFtLot`、`Bathrooms` 和 `Bedrooms` 等的回归系数都是负值。原始回归模型中并未包含表示地段的变量，而地段是房屋价格的一个重要预测变量。为了对地段情况建模，我们加入了变量 `ZipGroup`。该变量将邮政编码分到 5 个组中的一个，从房价最便宜的组 1 到房价最贵的组 5[6]。

```
lm(AdjSalePrice ~  SqFtTotLiving + SqFtLot +
     Bathrooms + Bedrooms +
     BldgGrade + PropertyType + ZipGroup,
   data=house, na.action=na.omit)
```

注 6：金县房屋数据中有 82 个邮政编码，但在部分邮政编码区域中，只有少量的房屋销售数据。另一种方法是直接使用邮政编码作为一个因子变量，`ZipGroup` 将相似的邮政编码聚为一个组。更多细节，参见 4.4.2 节。

```
Coefficients:
              (Intercept)              SqFtTotLiving
               -6.709e+05                  2.112e+02
                  SqFtLot                  Bathrooms
                4.692e-01                  5.537e+03
                 Bedrooms                  BldgGrade
               -4.139e+04                  9.893e+04
PropertyTypeSingle Family       PropertyTypeTownhouse
                2.113e+04                 -7.741e+04
                ZipGroup2                  ZipGroup3
                5.169e+04                  1.142e+05
                ZipGroup4                  ZipGroup5
                1.783e+05                  3.391e+05
```

显然，ZipGroup 是一个重要的变量。从输出中可以看到，对于房价最高的邮政编码组中的房屋，其预计销售价格更高，接近 34 万美元。SqFtLot 和 Bathrooms 的系数现在是正值，增加一间浴室会将房屋售价提高近 7500 美元。

Bedrooms 的系数依然为负值。尽管这不直观，却是房地产行业中的一个众所周知的现象。对于居住面积和浴室数目相同的房子，如果卧室更多、面积更小，售价会更低。

4.5.4　交互作用和主效应

统计学家喜欢区分**主效应**（或自变量）和主效应之间的**交互作用**。主效应一般指回归方程中的**预测变量**。如果在模型中只使用主效应，那么一个隐含的假设就是，预测变量与响应变量之间的关系是与其他预测变量无关的。该假设通常并不成立。

例如，对于 4.5.3 节中使用金县房屋数据所拟合的模型，其中的主效应包括 ZipCode 等多个变量。地段在房地产行业是决定一切的因素。很自然，我们可以假定房屋面积和销售价格之间的关系是依赖于地段的。在低租金地段建造的大面积房屋，其售价将不同于在昂贵地段上建造的大面积房屋。在 R 中，可以使用 * 操作符添加变量间的交互作用。下面的代码使用金县房屋数据拟合了 SqFtTotLiving 和 ZipGroup 间的交互作用。

```
lm(AdjSalePrice ~  SqFtTotLiving*ZipGroup + SqFtLot +
     Bathrooms + Bedrooms + BldgGrade + PropertyType,
   data=house, na.action=na.omit)

Coefficients:
              (Intercept)          SqFtTotLiving
               -4.919e+05              1.176e+02
                ZipGroup2              ZipGroup3
               -1.342e+04              2.254e+04
                ZipGroup4              ZipGroup5
                1.776e+04             -1.555e+05
                  SqFtLot              Bathrooms
                7.176e-01             -5.130e+03
                 Bedrooms              BldgGrade
               -4.181e+04              1.053e+05
PropertyTypeSingle Family  PropertyTypeTownhouse
                1.603e+04             -5.629e+04
```

```
SqFtTotLiving:ZipGroup2    SqFtTotLiving:ZipGroup3
              3.165e+01                  3.893e+01
SqFtTotLiving:ZipGroup4    SqFtTotLiving:ZipGroup5
              7.051e+01                  2.298e+02
```

在生成的模型中具有 4 个新项，即 `SqFtTotLiving:ZipGroup2`、`SqFtTotLiving:ZipGroup3` 等。

我们可以看到，地段变量和房屋估价之间具有很强的交互作用。落在房价最低 `ZipGroup` 中的房屋，其斜率与主效应 `SqFtTotLiving` 的斜率相同，都是每平方英尺 177 美元。这是由于 R 对因子变量使用了**参考**编码，参见 4.4 节。位于房价最高 `ZipGroup` 中的房屋，其斜率是主效应加上 `SqFtTotLiving:ZipGroup5`，即 177 + 230 = 447 美元 / 平方英尺。换句话说，对于房价最贵邮政编码组中的房屋，房屋面积每增加一平方英尺，预测售价的提高量将是房价最低邮政编码组的 2.7 倍左右。



具有交互项的模型选择

对于涉及多变量的问题，确定模型中应包含哪些交互项是一个挑战性的问题。通常采取的方法有以下几种。

- 对于某些问题，可以使用先验知识和直觉，指导模型中应包含哪些交互项。
- 使用逐步选择法（参见 4.2.4 节），筛选各种模型。
- 使用惩罚回归自动拟合大量可能的交互项。
- 也许最常用的方法是**树模型**，以及其衍生的**随机森林**和**梯度提升树**。这类模型能自动搜索最佳的交互项，参见 6.2 节。

本节要点

- 考虑到预测因子之间的相关性，在多元线性回归中，必须注意如何解释回归系数。
- 多重共线性可能导致拟合回归方程中存在数值不稳定的问题。
- 混淆变量是指在模型中遗漏的重要预测因子，它可以导致存在虚假关系的回归方程。
- 如果变量和响应之间存在相互依赖的关系，那么需要在两个变量间添加一个交互项。

4.6　检验假设：回归诊断

我们在研究中做探索性建模时，除了评估上面介绍的各个指标（参见 4.2.2 节）之外，还要采取多个步骤去评估模型与数据的拟合度。这些步骤大多基于残差分析，因为对残差的分析可以检验模型所基于的假设。这些步骤并不直接解决预测的准确性问题，但是它们可以为预测提供一些有用的见解。

主要术语

标准残差

残差除以残差的标准误差。

离群值

距离其他记录（或预测结果）很远的记录（或结果值）。

强影响值（influential value）

一个值或记录，其存在与否会使回归方程有很大差异。

杠杆

单个记录对回归方程的影响程度。

同义词：帽值（hat-value）

非正态残差

非正态分布的残差可能会导致一些对回归的技术需求失效。但在数据科学中，通常并不会关注该问题。

异方差性

在输出的部分范围中具有较高变异性的残差。这可能表明在回归方程中缺失了某个预测变量。

偏残差图

展示结果变量和单个预测变量之间关系的一种诊断图。

同义词：变量添加图（added variable plot）

4.6.1 离群值

一般来说，极端的值会远离其他大部分观测值，我们称其为**离群值**。正如在估计位置和变异性时需要对异常值做一些处理（参见 1.3 节和 1.4 节），离群值可能会导致回归模型出现问题。在回归中，离群值的真实值会极大地偏离预测值。通过查看**标准残差**，就可以检测离群值。标准残差等于残差除以残差的标准误差。

并没有统计学理论说明如何从非离群值中分离出离群值。人们通常采用一种独断的经验法则，即确定一个观察值与大部分数据偏离多远才能称为离群值。例如，在使用箱线图时，离群值是距离箱子边界上下过远的数据点（参见 1.5.1 节）。这里的“过远”指的是“超出 1.5 倍四分位数间距”。在回归中，通常使用标准残差作为确定一个记录是否应归类为离群值的度量。标准残差可以解释为“距离回归线的标准误差倍数”。

下面，我们使用邮政编码 98105 区域的所有金县房屋销售数据拟合一个回归模型。

```
house_98105 <- house[house$ZipCode == 98105,]
lm_98105 <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                 Bedrooms + BldgGrade, data=house_98105)
```

我们使用 rstandard 函数抽取出标准残差，并使用 order 函数获得最小残差。

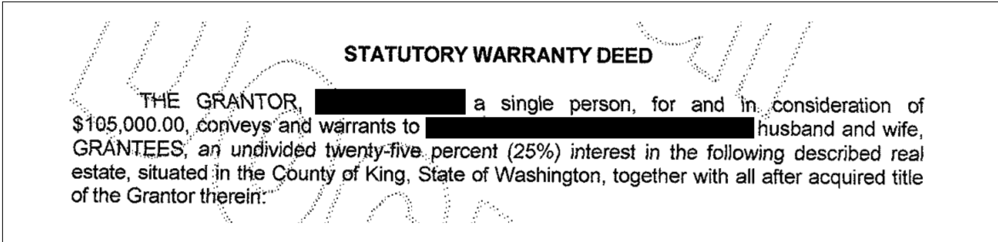
```
sresid <- rstandard(lm_98105)
idx <- order(sresid)
sresid[idx[1]]
    20431
-4.326732
```

在模型中，最大的过估计超出回归线之上四个标准误差，对应的过估计值是 757 753 美元。与该离群值对应的原始数据记录如下：

```
house_98105[idx[1], c('AdjSalePrice', 'SqFtTotLiving', 'SqFtLot',
              'Bathrooms', 'Bedrooms', 'BldgGrade')]

AdjSalePrice SqFtTotLiving SqFtLot Bathrooms Bedrooms BldgGrade
         (dbl)         (int)   (int)     (dbl)    (int)     (int)
1       119748          2900    7276         3        6         7
```

在本例中，记录看上去存在一些问题。在该邮政编码区域中，类似面积的房子一般售价远高于 119 748 美元。图 4-4 展示了本次售房法定契约的部分摘录。从图中可以看到，此次销售只涉及部分产权。因此，该离群值对应于一次异常的销售，不应该将该次销售包括在回归中。离群值也可能是由其他问题导致的，例如手工输入数据时出现了错误（即“胖手指”问题），或是单位不匹配，比如销售报告中的单位应是千美元，而非美元。



STATUTORY WARRANTY DEED

THE GRANTOR, [redacted] a single person, for and in consideration of $105,000.00, conveys and warrants to [redacted] husband and wife, GRANTEES, *an undivided twenty-five percent (25%) interest in the following described real* estate, situated in the County of King, State of Washington, together with all after acquired title of the Grantor therein:

图 4-4：具有最大负残差的售房记录对应的契约法定担保文书

在大数据问题中，对于拟合一个用于预测新数据的回归模型，通常离群值并不会构成问题。但是，离群值是异常检测所关注的核心问题。异常检测就是要找出数据中的离群值。离群值也可能对应于一次欺诈，或是一个意外操作。在任何情况下，检测离群值都可能是关键的业务需求。

4.6.2　强影响值

如果某个值的缺失会显著地改变回归方程，那么该值就是一个**强影响观测值**。在回归中，不需要将这样的值关联到大的残差。以图 4-5 的回归线为例。图中的实线对应于全部数据的回归，虚线对应于移除了右上方的数据点后的回归。很显然，对于使用所有数据的回归而言，尽管移除的数据点与大的离群点毫无关联，但是它对回归有很大的影响。我们称这样的数据点在回归中具有高**杠杆**。

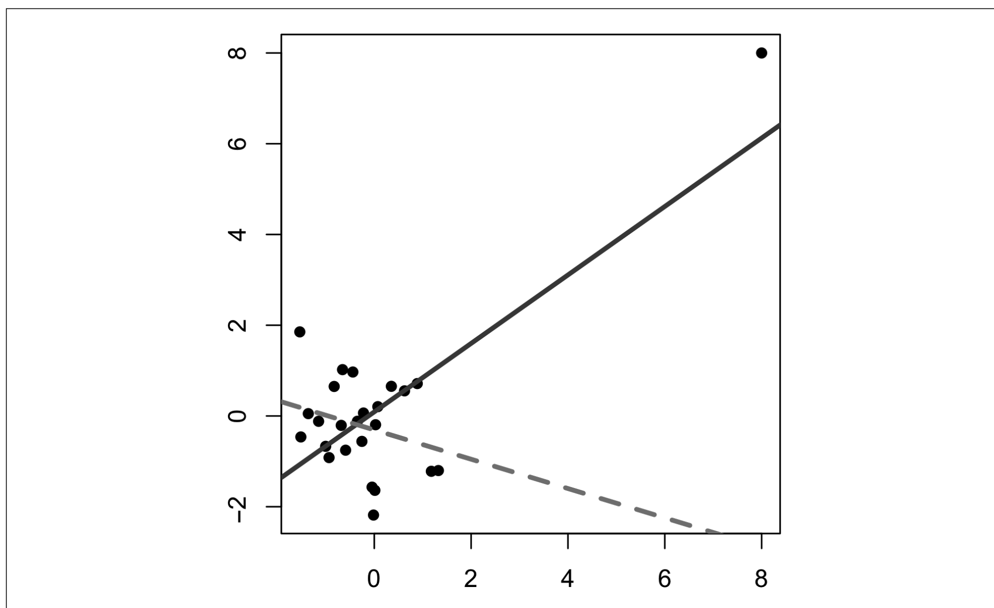


图 4-5：回归中的强影响数据点案例

为了确定单个记录对回归的影响，除了标准残差（参见 4.6.1 节）之外，统计学家还提出了多个度量。其中，**帽值**是对杠杆的一个常用度量。如果帽值高于 $2(P+1)/n$，表明存在一个高杠杆的数据值。[7]

另一个度量是**库克距离**（Cook's distance），它通过组合杠杆和残差规模，定义了对回归的影响情况。经验法则指出，如果库克距离大于 $4/(n-P-1)$，那么观测值具有很大的影响。

影响图，也被称为**气泡图**，在单个绘图中展示了标准残差、帽值和库克距离。图 4-6 显示了金县房屋数据的影响图，它是使用下面的 R 代码生成的。

```
std_resid <- rstandard(lm_98105)
cooks_D <- cooks.distance(lm_98105)
hat_values <- hatvalues(lm_98105)
plot(hat_values, std_resid, cex=10*sqrt(cooks_D))
abline(h=c(-2.5, 2.5), lty=2)
```

很明显，在回归中有多个数据点表现出了强影响。可以使用 `cooks.distance` 函数计算库克距离，使用 `hatvalues` 函数计算诊断信息。在图 4-6 中，x 轴表示帽值，y 轴表示残差，数据点的大小与库克距离相关。

注 7："帽值"一词来自回归中的**帽子矩阵**概念。多元线性回归可以表示为公式 $\hat{Y}=HY$，其中 H 是帽子矩阵。帽值对应于矩阵 H 的对角线。

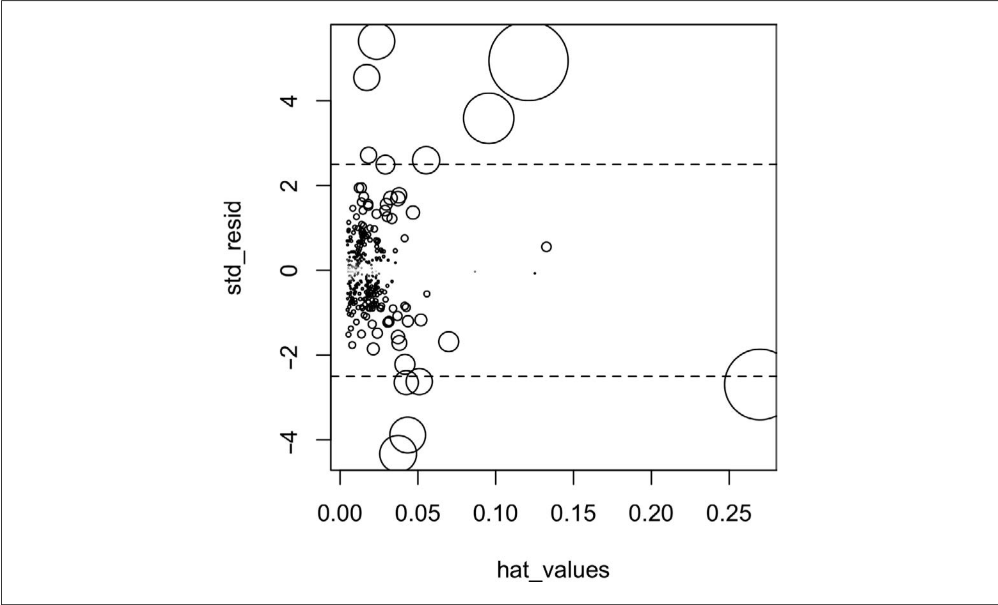


图 4-6：确定有最大影响的观测点的绘图

表 4-2 比较了回归与整个数据集，其中移除了强影响数据点。从表中可以看到，`Bathrooms` 回归系数的变化很大 [8]。

表4-2：使用全部数据以及移除强影响数据后，回归系数的对比情况

	原始数据	移除强影响数据后
截距	−772 550	−647 137
`SqFtTotLiving`	210	230
`SqFtLot`	39	33
`Bathrooms`	2282	−16 132
`Bedrooms`	−26 320	−22 888
`BldgGrade`	130 000	114 871

如果希望拟合的回归能可靠地预测未来的数据，那么识别强影响观测值只对小规模数据集有用。对于涉及大量记录的回归，单个观测值并不足以对拟合函数产生极端的影响（尽管回归可能依然有很大的离群值）。但是对于异常检测而言，识别强影响观测值是十分有用的。

4.6.3　异方差性、非正态分布和相关误差

统计学家十分关注残差的分布情况。普通最小二乘法（参见 4.1.3 节）已被证实是无偏的，并且在一些情况下，对于很多分布假设来说是一种最优的估计。这意味着，在多数问题

注 8：`Bathrooms` 的回归系数会变成负值，这是不直观的。这是因为回归中并未考虑地段的情况，并且在邮政编码 98105 区域中，还包括了不同类型的房屋。对混淆变量的讨论，参见 4.5.3 节。

中，数据科学家无须过于关心残差的分布情况。

残差分布主要与形式统计推断的有效性（即假设验证和 p 值）相关，这对于数据科学家而言是无关紧要的，因为数据科学家主要关心的是准确性。要使形式推断完全有效，需要假定残差符合正态分布，方差相同，并且是独立的。预测值置信区间的标准计算方法是数据科学家可能会关注的一个领域，这基于对残差的假设（参见 4.3.2 节）。

异方差性指在整个预测值范围内并不具有一个恒定的残差变异性。换句话说，在整个预测值范围内，部分数据的误差要大于其他部分的误差。`ggplot2` 软件包提供了一些分析残差的便利工具。

下面的代码使用了 4.6.1 节中的回归拟合的 `lm_98105` 模型，绘制了残差绝对值与预测值的对比情况。

```
df <- data.frame(
  resid = residuals(lm_98105),
  pred = predict(lm_98105))
ggplot(df, aes(pred, abs(resid))) +
  geom_point() +
  geom_smooth()
```

绘图如图 4-7 所示。使用 `geom_smooth` 函数，很容易实现残差绝对值的平滑叠加。该函数调用了 `loess` 方法，对散点图中 x 轴和 y 轴变量之间的关系生成了一种可视化平滑（参见本节后面的“散点图平滑”）。

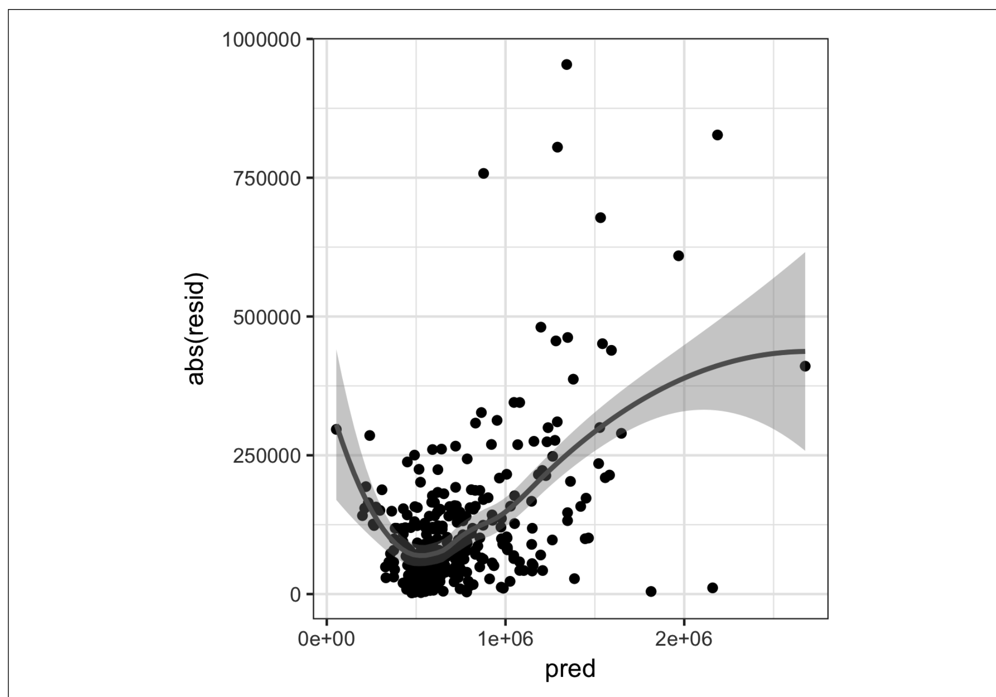
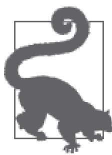


图 4-7：残差绝对值和预测值

很明显，对于高售价的房屋，残差的方差趋向于增大；但是对于低售价的房屋，残差的方差也趋向于增大。绘图显示，模型 `lm_98105` 的误差具有**异方差性**。



为什么数据科学家要关注异方差性？

异方差性表明在不同的预测值范围内，预测误差存在差异，还表明模型可能并不完整。例如，模型 `lm_98105` 的异方差性表明，在回归中可能并未统计在高售价范围和低售价范围内的一些房屋。

图 4-8 显示了模型 `lm_98105` 回归标准残差的直方图。其分布比正态分布具有更长的尾部，并略向更大的残差偏斜。

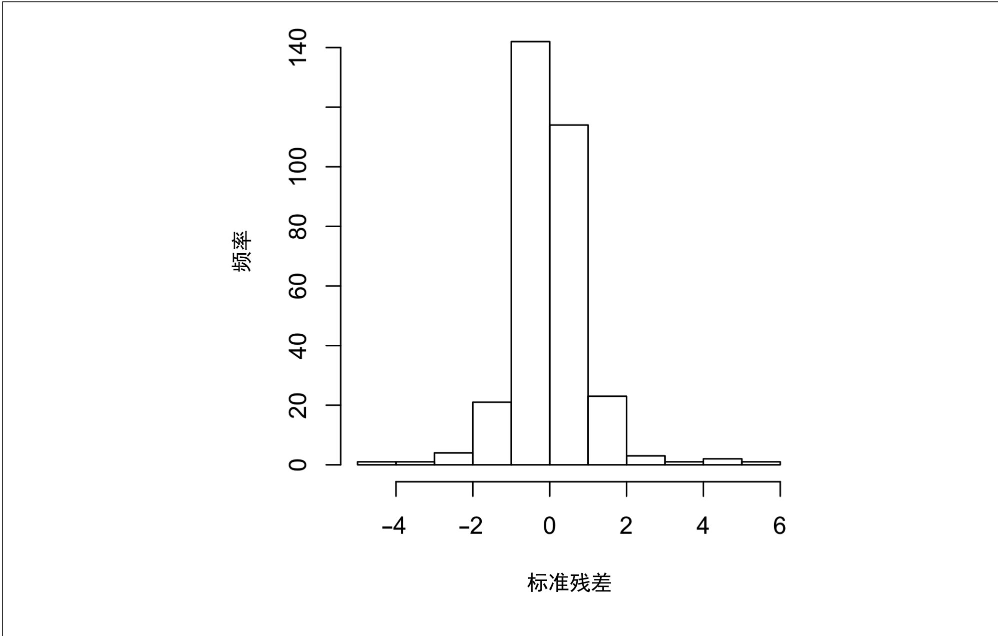


图 4-8：金县房屋数据回归残差的直方图

统计学家可能还会检验误差是独立的这一假设。对于在一段时间内采集的数据，该假设尤为正确。Durbin-Watson **统计量**可用于检测在涉及时序数据的回归中，是否存在显著的自相关。

尽管在回归中可能会违反其中一种分布假设，但是数据科学家为什么要关心这个问题？在数据科学中，通常主要关注的是预测的准确性，因此审视一下异方差性可能会有所帮助。我们可能会发现，数据中有些信号未被模型捕获。满足正态分布的假设，仅是为了验证形式统计推断（p 值、F 统计量等），对于数据科学家则无关紧要。



散点图平滑

回归就是建模响应变量和预测变量之间的关系。在评估一个回归模型时，使用**散点图平滑**以可视化方式明确两个变量之间的关系是有用的。

以图 4-7 为例，对绝对残差和预测值之间关系的平滑，显示了残差的方差依赖于残差的值。在平滑中，使用了 `loess` 函数。`loess` 函数重复地对邻近子集拟合出一系列本地回归，进而实现平滑。尽管 `loess` 函数可能是最广为使用的一种平滑函数，但是 R 中还提供了其他一些散点图平滑函数，例如超平滑（`supsmu`）和核平滑（`ksmooth`）。如果想要评估一个回归模型，通常并不需要关心这些散点图平滑的工作细节。

4.6.4　偏残差图和非线性

偏残差图以可视化方式展示了估计的拟合值是否很好地反映了预测变量和输出之间的关系。偏残差图和离群值检测是数据科学家最重要的诊断手段。偏残差图的基本理念是，将预测变量与响应变量间的关系独立出来，**并考虑所有其他的预测变量**。偏残差可以看成一种“合成的输出值”，其中组合了基于单个预测变量的预测值，以及来自完全回归方程的实际残差。预测变量 X_i 的偏残差是普通残差加上与 X_i 关联的回归项。

$$偏残差 = 残差 + \hat{b}_i X_i$$

其中，$\hat{b}_i$ 是估计的回归系数。R 中的 `predict` 函数提供了返回单个回归项 $\hat{b}_i X_i$ 的选项。

```
terms <- predict(lm_98105, type='terms')
partial_resid <- resid(lm_98105) + terms
```

偏残差图在 x 轴上显示 X_i，在 y 轴上显示偏残差。使用 `ggplot2`，很容易实现在已有绘图上叠加偏残差的平滑绘图，代码如下：

```
df <- data.frame(SqFtTotLiving = house_98105[, 'SqFtTotLiving'],
                 Terms = terms[, 'SqFtTotLiving'],
                 PartialResid = partial_resid[, 'SqFtTotLiving'])
ggplot(df, aes(SqFtTotLiving, PartialResid)) +
  geom_point(shape=1) + scale_shape(solid = FALSE) +
  geom_smooth(linetype=2) +
  geom_line(aes(SqFtTotLiving, Terms))
```

绘图结果如图 4-9 所示。偏残差是对 `SqFtTotLiving` 添加到价格中的贡献的估计。显然，`SqFtTotLiving` 与房屋售价间的关系是非线性的。回归线低估了面积小于 1000 平方英尺的房屋的售价，高估了面积在 2000 到 3000 平方英尺的房屋的售价。鉴于房屋面积大于 4000 平方英尺的数据点太少，因此难以对这些房屋给出结论。

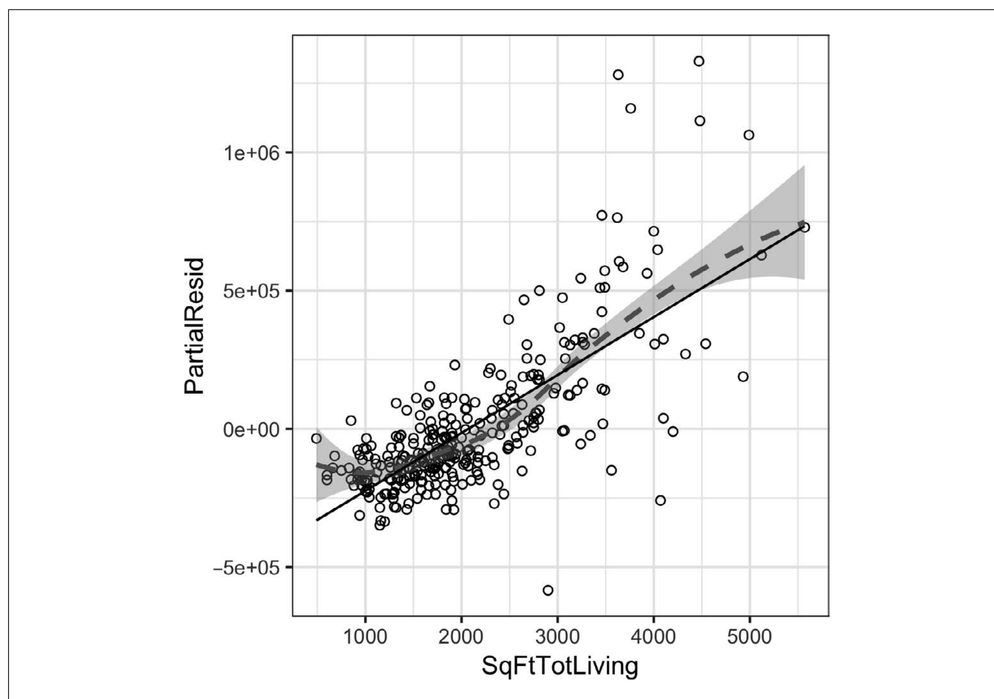


图 4-9：变量 SqFtTotLiving 的偏残差图

在本例中，非线性是有意义的。房屋面积同样增大 500 平方英尺，小型房屋的售价会比大型房屋有更大的变化。这表明对于 SqFtToLiving，不应只考虑简单的线性项，还应考虑一些非线性项（参见 4.7 节）。

本节要点

- 鉴于离群值可能会在小规模数据集中导致问题，关注离群值主要是为了发现数据中存在的问题，或是确定异常所在。
- 单个记录（包括回归离群值）可以对小规模数据集的回归方程产生很大的影响。但是在大数据中，这种效果却荡然无存。
- 如果将回归模型用于形式推断（如 p 值等），那么应该检验对残差分布的一些假设。但是对于数据科学而言，残差分布通常无关紧要。
- 偏残差图可以用于定性地评估每个回归项的拟合情况，这可能会得出另一种模型声明。

4.7 多项式回归和样条回归

响应变量和预测变量之间的关系并非总是线性的。例如，药物剂量的响应变量通常就不是线性的：剂量加倍一般不会导致响应加倍。产品需求也不是营销投入的线性方程，因为需求总会在某一点上饱和。扩展回归以捕获这些非线性效果的方法有多种。

主要术语

多项式回归

在回归方程中添加了多项式项，例如平方项、三次方项等。

样条回归

使用一系列多项式片段去拟合一条平滑曲线。

结点

分隔样条片段的值。

广义加性模型

可以自动选择结点的样条模型。

同义词：GAM



非线性回归

统计学家所说的**非线性回归**，指的是那些不能使用最小二乘法拟合的模型。那么，哪些类型的模型是非线性的？从本质上讲，所有响应不能表示为预测变量（或预测变量的某种转换）的线性组合的模型，都是非线性的。非线性回归模型需要做数值优化，因此更难以拟合，计算的强度也更大。如有可能，我们应尽量使用线性模型。

4.7.1 多项式回归

多项式回归涉及在回归方程中添加多项式项。多项式回归的使用可以追溯至葛尔刚（Gergonne）在 1815 年的论文提出了回归。例如，响应变量 Y 和预测变量 X 间的二项式回归的形式如下。

$$Y = b_0 + b_1 X + b_2 X^2 + e$$

可以使用 R 中的 `poly` 函数拟合多项式回归。例如，下面的代码使用金县房屋数据，对 `SqFtTotLiving` 拟合了一个二项式回归。

```
lm(AdjSalePrice ~  poly(SqFtTotLiving, 2) + SqFtLot +
                BldgGrade +  Bathrooms +  Bedrooms,
                    data=house_98105)

Call:
lm(formula = AdjSalePrice ~ poly(SqFtTotLiving, 2) + SqFtLot +
    BldgGrade + Bathrooms + Bedrooms, data = house_98105)

Coefficients:
            (Intercept)  poly(SqFtTotLiving, 2)1
             -402530.47               3271519.49
poly(SqFtTotLiving, 2)2                  SqFtLot
              776934.02                    32.56
```

```
BldgGrade               Bathrooms
135717.06                -1435.12
 Bedrooms
 -9191.94
```

这里 SqFtTotLiving 关联了两个回归系数，一个用于线性项，另一个用于平方项。

图 4-10 显示了拟合的偏残差图（参见 4.6.4 节）。图中表明，在关联 SqFtTotLiving 的回归方程中存在一个曲率。相比于线性拟合，拟合线更接近对匹配偏残差的平滑（参见 4.7.2 节）。

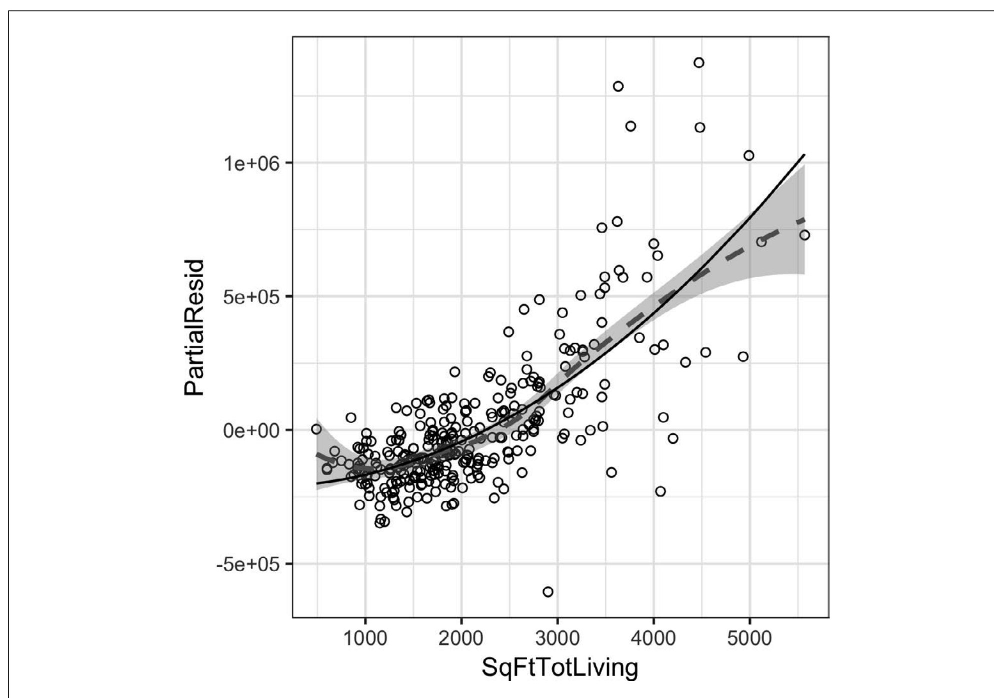


图 4-10：变量 SqFtTotLiving 的多项式回归拟合（实线），以及与平滑（虚线，参见 4.7.2 节中对样条回归的介绍）的对比

4.7.2　样条回归

多项式回归只捕获了非线性关系的部分曲率。添加高阶项（例如三次方项），通常会导致回归线中出现我们所不期望的“摇摆”（wiggliness）现象。还有一种方法，通常也是更好的做法，是在建模非线性关系时使用**样条**。样条是一种在不动点间平滑插值的方法。样条最初是手工业者在绘制平滑曲线时所使用的工具，特别是在轮船和飞机制造中。

样条是通过使用重物［俗称“鸭子”（ducks）］弯曲一根细木条得到的，如图 4-11 所示。



图 4-11：样条最初是使用可弯曲的木条和“鸭子”构建的，它是手工业者拟合曲线的一种工具。Bob Perry 摄

从技术上定义，样条是一组分段的连续多项式。样条函数是第二次世界大战期间，由罗马尼亚数学家艾萨克·雅各布·勋伯格在美国阿伯丁试验场首次提出的。多项式片段在预测变量中的一组不动点处平滑地连接，这些不动点被称为**结点**。相比于多项式回归，样条函数的计算公式要复杂得多。样条函数的拟合细节通常由统计软件处理。R 的 `splines` 软件包就提供了 `bs` 函数，可以在回归模型中创建一个 *b* **样条**项。例如，下面的代码在金县房屋回归模型中添加了一个 *b* 样条项。

```
library(splines)
knots <- quantile(house_98105$SqFtTotLiving, p=c(.25, .5, .75))
lm_spline <- lm(AdjSalePrice ~ bs(SqFtTotLiving, knots=knots, degree=3) +
  SqFtLot + Bathrooms + Bedrooms + BldgGrade,  data=house_98105)
```

使用 `bs` 函数时需要指定两个参数：多项式的幂次数和结点的位置。在本例中，添加到模型中的预测变量 `SqFtTotLiving` 使用了三次样条（`degree=3`）。在默认情况下，`bs` 函数会将结点置于各个边界处。此外，结点也可置于下四分位数、中四分位数和上四分位数等处。

线性项的回归系数具有直接的意义，但样条项的系数是不可解释的。以可视化方式揭示样条拟合的本质更加有用。图 4-12 展示了回归给出的偏残差图。相比于多项式模型，样条模型更近似于匹配了平滑，这表明样条具有更大的灵活性。在本例中，线条更近乎于拟合了数据。这是否意味着样条回归是一种更好的模型？不一定。在本例中，我们可以看到，面积非常小的房屋（小于 1000 平方英尺）的售价将比面积稍大的房屋更高，显然这并不符合经济规律。问题可能是由于混淆变量导致的，参见 4.5.3 节。

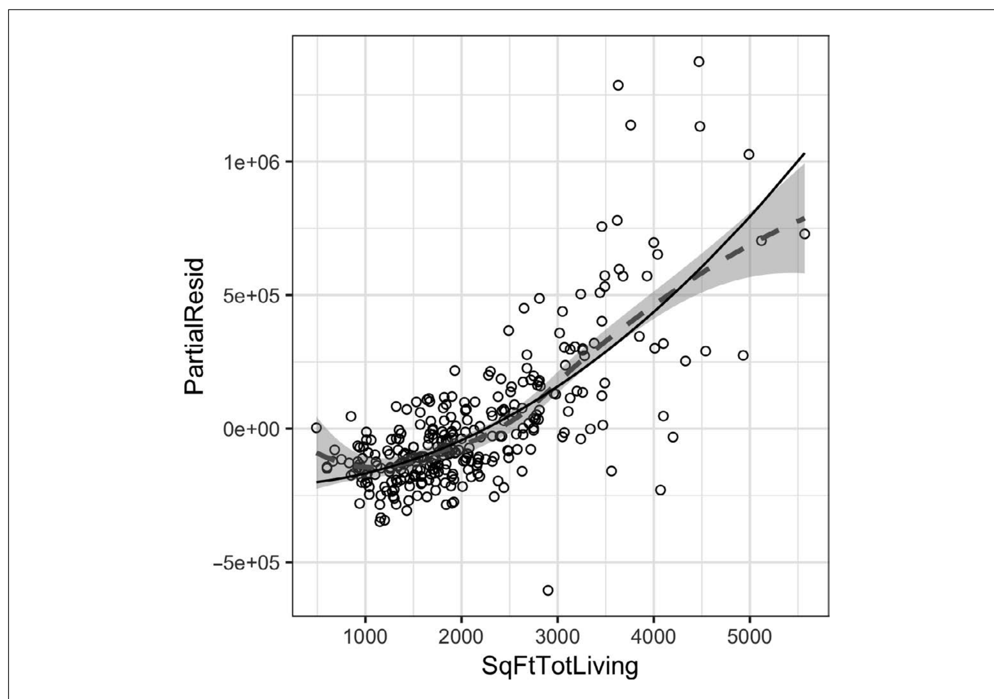


图 4-12：变量 SqFtTotLiving 的样条回归拟合（实线）与平滑（虚线）的对比

4.7.3 广义加性模型

假设我们基于先验知识或回归诊断，怀疑响应变量和预测变量之间存在某种非线性关系。多项式项可能不够灵活，无法捕获这种非线性关系，而样条项则需要指定各个结点。**广义加性模型**（GAM）是一种自动拟合样条回归的方法。可使用 R 的 gam 软件包，拟合金县房屋数据的广义加性模型。

```
library(mgcv)
lm_gam <- gam(AdjSalePrice ~ s(SqFtTotLiving) + SqFtLot +
                      Bathrooms +  Bedrooms + BldgGrade,
                    data=house_98105)
```

其中，s(SqFtTotLiving) 项告诉 gam 函数为样条项找出“最好”的结点，如图 4-13 所示。

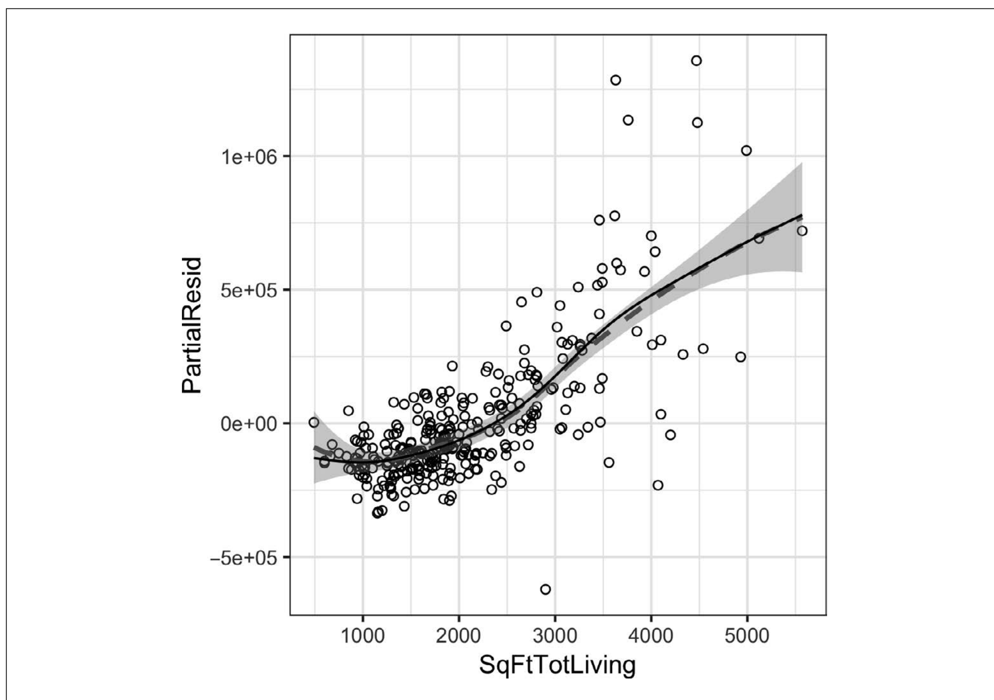


图 4-13：变量 SqFtTotLiving 的广义加性模型回归拟合（实线）与平滑（虚线）的对比

本节要点

- 在回归中，离群值表现为具有很大残差的记录。
- 多重共线性会导致拟合回归方程中存在数值不稳定的问题。
- 混淆变量是一种重要的预测变量。如果在一个模型中忽略了混淆变量，将会导致回归方程给出伪关系。
- 如果一个变量的效果依赖于另一个变量（因子变量）的层级，那么在两个变量之间需要有交互项。
- 多项式回归可以拟合预测变量和结果变量之间的非线性关系。
- 样条是一组连接在一起的多项式片段，连接点被称为结点。
- 广义加性模型可以自动指定样条函数中的结点。

4.7.4　拓展阅读

关于样条模型和广义加性模型的更多内容，可参见 Trevor Hastie、Robert Tibshirani 和 Jerome Friedman 合著的《统计学习基础：数据挖掘、推理与预测》，以及《统计学习导论：基于 R 应用》，该书由 Gareth James、Daniela Witten、Trevor Hastie 和 Robert Tibshirani 合著。两本书均由 Springer 出版社出版。

4.8 小结

多年来我们已经看到，回归的应用比其他任何一种统计方法都广泛。回归是建立多个预测变量与一个结果变量之间关系的过程。回归的基础形式是线性回归，即每个预测变量具有一个回归系数，它描述了预测变量和结果变量之间的线性关系。在一些更高级的回归中，例如在多项式回归和样条回归中，回归关系可以是非线性的。经典统计学的重点在于发现对观测数据的良好拟合，以解释或描述一些现象。使用传统（即“样本内”）的度量去评估一个模型，这是拟合的强大之处。与之不同，数据科学的目标通常是预测新数据的值，因此使用的是基于对“样本外”数据预测准确性的度量。采用变量选择方法来降维，创建更紧致的模型。

第 5 章

分类

数据科学家经常会遇到需要自动做出决策的问题。例如，一封电子邮件是否试图进行钓鱼式攻击？一名客户是否会流失？一位网络用户是否会点击一个广告？这些问题都可以归为**分类**问题。分类可能是最重要的预测形式，其目标是预测一个记录值是 0 还是 1（钓鱼式攻击或非钓鱼式攻击，客户会流失或者不会流失，用户会点击或者不会点击），或者在某些情况下是预测类别之一（例如，Gmail 就将收件箱中的邮件分为“主要”“社交”“促销”和“论坛”类）。

在很多情况下，我们需要的不仅仅是简单的二分类，还需要预测一个实例属于某个类的概率。

大多数算法并非只是让模型简单地指定一个二分类，而是可以返回属于感兴趣的类的概率（或倾向）。事实上，在使用 R 实现逻辑回归时，默认输出是对数几率形式的，必须转换为倾向分值，然后才能使用一个滑动的截止值将倾向分值转换为决策。通用方法如下。

(1) 对感兴趣的类确定一个截止概率值。超过此值，就可以认为记录属于该类。
(2) 估计任何模型中一个记录属于我们感兴趣的类的概率。
(3) 如果该概率值大于所确定的截止概率值，那么可以将新记录指定为感兴趣的类。

截止概率越高，被预测为 1（即属于感兴趣的类）的记录越少。反之，截止概率越低，被预测为 1 的记录越多。

本章将介绍一些用于分类和估计倾向性的重要方法。更多关于分类和数值预测的方法，将在下一章中介绍。

两个以上的类别？

绝大多数问题只涉及二元响应变量。但是在一些分类问题中，响应变量有两个以上的结果。例如，当客户的订阅合同满一年后，可能会有三个结果：客户离开或“流失”（$Y = 2$），改为按月订阅合同（$Y = 1$），签署新的长期合同（$Y = 0$）。我们的目标是预测 $Y = j$，其中 $j = 0$、1 或 2。本章介绍的大部分分类方法，都可以直接或经过小的修改后用于具有两个以上结果的响应变量。即使是在具有两个以上结果的情况下，通常也可以使用条件概率，将问题重写为一系列二分类问题。例如，为了预测订阅合同的结果，我们可以将该问题转换为两个二分类预测问题。

- 预测 $Y = 0$ 还是 $Y > 0$。
- 如果 $Y > 0$，预测 $Y = 1$ 还是 $Y = 2$。

将该问题分解为两种情况是一种合理的做法。首先判断客户是否会流失。如果客户没有流失，那么判断客户将会选择哪种类型的合同。从模型拟合的角度来看，通常最好将多分类问题转换为一系列二分类问题，尤其是在一个类别比其他类别更为普遍的情况下。

5.1 朴素贝叶斯算法

朴素贝叶斯算法使用在给定输出情况下观测到预测因子值的概率，估计给定一组预测因子的值时观测到结果 $Y = i$ 的概率。[1]

主要术语

条件概率

在给定另一个事件（比如 $Y = i$）的条件下，观测到某个事件（比如 $X = i$）的概率，记作 $P(X_i \mid Y_i)$。

后验概率

在给定预测因子的情况下，出现某一结果的概率（后验概率不同于结果的**先验概率**，后者并未考虑预测因子的信息）。

为了理解贝叶斯分类，我们从设想“非朴素”的贝叶斯分类开始。对于每个要分类的记录：

(1) 找到其他所有具有相同预测因子（即预测因子的值相同）的记录；
(2) 确定这些记录所属的类，以及其中哪个类是主要类（即最有可能的类）；
(3) 将该类指定给新记录。

使用上面的方法，可以找出样本中与要分类的新记录完全相同（指所有预测因子值相同）的所有记录。

注 1：本章中各节的内容，版权属于本书作者彼得·布鲁斯和安德鲁·布鲁斯，© 2017 Datastats，LLC。使用需经许可。



在标准的朴素贝叶斯算法中，预测因子必须是分类（因子）变量。5.1.3 节将介绍两种适用于连续变量的变通方法。

5.1.1 准确的贝叶斯分类是不切实际的

如果预测变量超出一定的数量，那么很多待分类的记录就无法准确地匹配。为了解释这个问题，下面我们以基于人口统计变量的投票预测模型为例。即使样本的规模相当大，很有可能其中也不会包含能匹配下列条件的新记录："来自美国中西部地区的西班牙裔男性美国人，并且具有高收入，在上次选举中投了票，在之前的选举中没有投票，有 3 个女儿和一个儿子，已经离婚。"在本例中只有 8 个变量，但是对于大多数分类问题而言，变量数目远多于此。只是在 5 个同等频繁出现的类别中添加了一个新变量，就将匹配的概率降低至原来的 20%。



尽管我们称该方法为"朴素贝叶斯"，但不应将其当作一种贝叶斯统计方法。朴素贝叶斯是一种数据驱动的经验性方法，仅需要具备一点点统计学专业知识。它的名字源于与**贝叶斯规则**类似的且用于预测的计算方式，更详细地说，就是在给定结果的情况下对预测值概率的初始计算，以及对结果概率的最终计算。

5.1.2 朴素解决方案

在朴素贝叶斯解决方案中，我们不再将概率计算局限于那些匹配待分类记录的记录上，而是使用整个数据集。朴素贝叶斯的改进如下。

(1) 对于二元响应变量 $Y = i$（$i = 0$ 或 1），估计每个预测因子的条件概率 $P(X_j \mid Y = i)$。这些是当观测到 $Y = i$ 时，预测因子值在记录中的概率。概率估计值为训练集中 $Y = i$ 的记录中 X_j 值的比例。
(2) 将这些概率相乘，再乘以属于 $Y = i$ 的记录的比例。
(3) 对所有的类，重复步骤 1 和步骤 2。
(4) 将步骤 2 中对类 i 计算得到的值，除以对所有类计算得到的这些值的总和，得到对结果 i 的概率估计。
(5) 将记录指派给对于该组预测值具有最大概率的类。

朴素贝叶斯算法也可以定义为，在给定一组预测值 $X_1, \cdots, X_p$ 的情况下，观测到结果 $Y = i$ 的概率：

$$P(X_1, X_2, \cdots, X_p)$$

概率值 $P(X_1, X_2, \cdots, X_p)$ 是一个比例因子，它可确保概率值介于 0 和 1 之间，并且不依赖于 Y。

$$P(X_1, X_2, \cdots, X_p) = P(Y = 0)(P(X_1 \mid Y = 0)P(X_2 \mid Y = 0) \cdots P(X_p \mid Y = 0)) + \\ P(Y = 1)(P(X_1 \mid Y = 1)P(X_2 \mid Y = 1) \cdots P(X_p \mid Y = 1))$$

为什么称上面的公式是“朴素的”（naive）？这是因为我们做了一个简单的假设，即在给定观测结果的情况下，预测因子值向量的**确切条件概率**，可以由单个条件概率 $P(X_j \mid Y = i)$ 的乘积很好地估计出来。换句话说，我们假定 X_j **独立于**其他所有的预测变量 X_k（$k \neq j$），这样就可以对 $P(X_j \mid Y = i)$ 做出估计，而非 $P(X_1, X_2, \cdots, X_p \mid Y = i)$。

可使用一些 R 包估计朴素贝叶斯模型。下面的代码就使用了 klaR 软件包去拟合模型。

```
library(klaR)
naive_model <- NaiveBayes(outcome ~ purpose_ + home_ + emp_len_,
                          data = na.omit(loan_data))
naive_model$table
$purpose_
          var
grouping   credit_card debt_consolidation home_improvement major_purchase
  paid off   0.1857711          0.5523427       0.07153354     0.05541148
  default    0.1517548          0.5777144       0.05956086     0.03708506
          var
grouping      medical      other small_business
  paid off 0.01236169 0.09958506     0.02299447
  default  0.01434993 0.11415111     0.04538382

$home_
          var
grouping    MORTGAGE        OWN      RENT
  paid off 0.4966286 0.08043741 0.4229340
  default  0.4327455 0.08363589 0.4836186

$emp_len_
          var
grouping    > 1 Year   < 1 Year
  paid off 0.9690526 0.03094744
  default  0.9523686 0.04763140
```

模型的输出是条件概率 $P(X_j \mid Y = i)$。可以使用该模型预测一笔新贷款的结果。

```
new_loan
        purpose_    home_  emp_len_
1 small_business MORTGAGE  > 1 Year
```

在本例中，模型预测了一次贷款拖欠。

```
predict(naive_model, new_loan)
$class
[1] default
Levels: paid off default

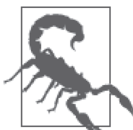
$posterior
      paid off   default
[1,] 0.3717206 0.6282794
```

预测还返回了贷款拖欠概率 posterior 的估计值。我们知道，朴素贝叶斯分类会生成**有偏**估计。然而，如果我们的目标是根据 $Y = 1$ 的概率值对记录**排序**，那么就不需要概率的无偏估计，朴素贝叶斯就能给出很好的结果。

5.1.3 数值型预测变量

从定义中可以看出，贝叶斯分类器仅适用于分类预测变量。例如，在垃圾邮件分类中，预测任务关注的是邮件中是否存在某个单词、短语或字符等。要将朴素贝叶斯用于数值型预测变量，需要采取下面两种方法之一。

- 将数值型预测变量划分为多个箱子，并转换为分类预测因子，然后再应用上面介绍的算法。
- 使用正态分布（参见 2.6 节）等概率模型，估计条件概率 $P(X_j \mid Y = i)$。



如果训练数据中不存在预测因子类，那么在新数据中，算法会对结果变量赋予**零概率**。而其他一些方法会直接忽略该变量，并使用其他变量给出的信息。在对连续变量分箱时，需要注意这一点。

本节要点

- 朴素贝叶斯适用于分类的（因子型的）预测和结果。
- 朴素贝叶斯要解答的问题是："在每个结果类别中，哪些预测类别是最可能发生的？"
- 该问题可以转化为，在给定预测值的情况下，估计结果属于不同类别的概率。

5.1.4 拓展阅读

- Trevor Hastie、Robert Tibshirani 和 Jerome Friedman 合著的《统计学习基础（第 2 版）》。
- 在 Galit Shmueli、Peter Bruce 和 Nitin Patel 合著的 *Data Mining for Business Analytics: Concepts, Techniques, and Applications with XLMiner* 一书中，有一整章的内容介绍朴素贝叶斯。该书有针对 R、Excel 和 JMP 的不同版本。

5.2 判别分析

判别分析是最早提出的统计分类器。1936 年，统计学家 R. A. Fisher 于在 *Annals of Eugenics* 期刊上发表了一篇文章，首次提出了这一概念。[2]

主要术语

协方差

对一个变量相对于另一个变量的一致程度（幅度和方向类似）的度量。

判别函数

当应用于预测变量上时，该函数可以使类之间的分离度最大化。

判别权重

应用判别函数得到的分值，用于估计记录属于某个类的概率。

注 2：确实令人惊讶的是，第一篇关于统计分类的文章，竟然发表在专门针对优生学研究的期刊上。事实上，在早期发展中，统计学的确与优生学密切相关。

判别分析包含了很多种方法，其中最常用的是**线性判别分析法**（LDA）。事实上，费希尔提出的原始方法与线性判别分析法略有差异，但原理基本相同。随着树模型和逻辑回归等更复杂的方法的提出，如今线性判别分析法的使用不再那么广泛了。

但是，我们依然能在某些应用中遇到线性判别分析法。此时，线性判别分析法可能关联使用了其他更广为使用的方法，例如主成分分析（参见 7.1 节）等。此外，判别分析可以提供对预测因子重要性的度量，并且在特征选择上也是一种计算效率很高的方法。



隐含狄利克雷分布（Latent Dirichlet Allocation）同样被简写为 LDA，但不要将它和线性判别分析法混淆了。隐含狄利克雷分布主要用于文本和自然语言处理，与线性判别分析毫无关系。

5.2.1 协方差矩阵

为了理解判别分析，我们先介绍一下两个或多个变量间**协方差**的概念。协方差衡量了两个变量 x 和 z 之间的关系。如果用 $\bar{x}$ 和 $\bar{z}$ 分别表示变量 x 和 z 的均值（参见 1.3.1 节），那么 x 和 z 间的协方差 $s_{x,z}$ 可由下式给出。

$$s_{x,z}=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(z_i-\bar{z})}{n-1}$$

其中，n 是记录的个数。注意，公式中的除数是 $n-1$，而不是 n。参见 1.4.1 节中的知识点“自由度是 n，还是 $n-1$？”。

和相关系数（参见 1.7 节）一样，协方差为正值表示正相关，为负值表示负相关。但是，相关系数的值限定在 −1 到 1 的区间内，而协方差与变量 x 和 z 具有相同的尺度。在 x 和 z 的**协方差矩阵** Σ 中，对角线元素（即行和列对应于同一变量）为单个变量的方差，即 s_x^2 和 s_z^2，而非对角线元素是相应变量对间的协方差。

$$\Sigma=\begin{bmatrix} s_x^2 & s_{x,z} \\ s_{x,z} & s_z^2 \end{bmatrix}$$



回想一下，标准偏差用于将变量归一化为 z 分数。协方差矩阵是对多变量扩展的归一化过程。这种归一化被称为**马氏距离**（参见 6.1.2 节中的知识点“其他距离度量”），而且它与线性判别分析函数相关。

5.2.2 费希尔线性判别分析

为了简单起见，我们侧重于其中一种分类问题，即使用两个连续的数值变量 (x, z) 预测二元结果 y。从技术角度上看，判别分析假设预测变量是符合正态分布的连续变量，但在实践中，该方法也适用于与正态分布偏离不大的情况，也适用于二元预测因子。费希尔线性判别法区分了**组间**变异性和**组内**变异性。具体而言，在将记录划分为两组时，线性判别

分析法侧重于相对“组内”平方和 SS_{within}（测量了组内的变异性）最大化“组间”平方和 $SS_{between}$（测量了组间的变异性）。在这种情况下，两个组分别对应于 $y = 0$ 时的记录 (x_0, z_0) 和 $y = 1$ 时的记录 (x_1, z_1)。该方法找出最大化平方和比率 $\frac{SS_{between}}{SS_{within}}$ 的线性组合 $w_x x + w_z z$。组间平方和是两个组的均值之间距离的平方，而组内平方和是围绕组内均值的分散程度，并以协方差矩阵为权重。直观而言，该方法通过最大化组间平方和并最小化组内平方和，生成两组之间的最大分离。

5.2.3 一个简单的例子

W. N. Venables 和 B. D. Ripley 在其合著的 *Modern Applied Statistics With S* 一书中介绍了 MASS 软件包。MASS 软件包提供了实现线性判别分析法的 R 函数。下面，我们使用两个预测变量 loaner_score 和 payment_inc_ratio 在贷款数据的一个样本上应用该函数，并给出线性判别器权重的估计值。

```
library(MASS)
loan_lda <- lda(outcome ~ borrower_score + payment_inc_ratio,
                     data=loan3000)
loan_lda$scaling
                          LD1
borrower_score    -6.2962811
payment_inc_ratio  0.1288243
```



在特征选择中使用判别分析法

如果预测变量在使用线性判别分析法之前已被归一化，那么判别器的权重测定了变量的重要性，这为特征选择提供了一种计算效率很高的方法。

lda 函数可以预测贷款的“拖欠”（default）与“付清”（paid off）概率。

```
pred <- predict(loan_lda)
head(pred$posterior)
       paid off   default
25333 0.5554293 0.4445707
27041 0.6274352 0.3725648
7398  0.4014055 0.5985945
35625 0.3411242 0.6588758
17058 0.6081592 0.3918408
2986  0.6733245 0.3266755
```

为了进一步阐明线性判别分析法的工作原理，我们对预测情况进行绘图。我们使用预测函数 lda 的输出，估计一个拖欠概率的曲线。

```
lda_df <- cbind(loan3000, prob_default=pred$posterior[,'default'])
ggplot(data=lda_df,
       aes(x=borrower_score, y=payment_inc_ratio, color=prob_default)) +
  geom_point(alpha=.6) +
  scale_color_gradient2(low='white', high='blue') +
geom_line(data=lda_df0, col='green', size=2, alpha=.8) +
```

生成的绘图如图 5-1 所示。

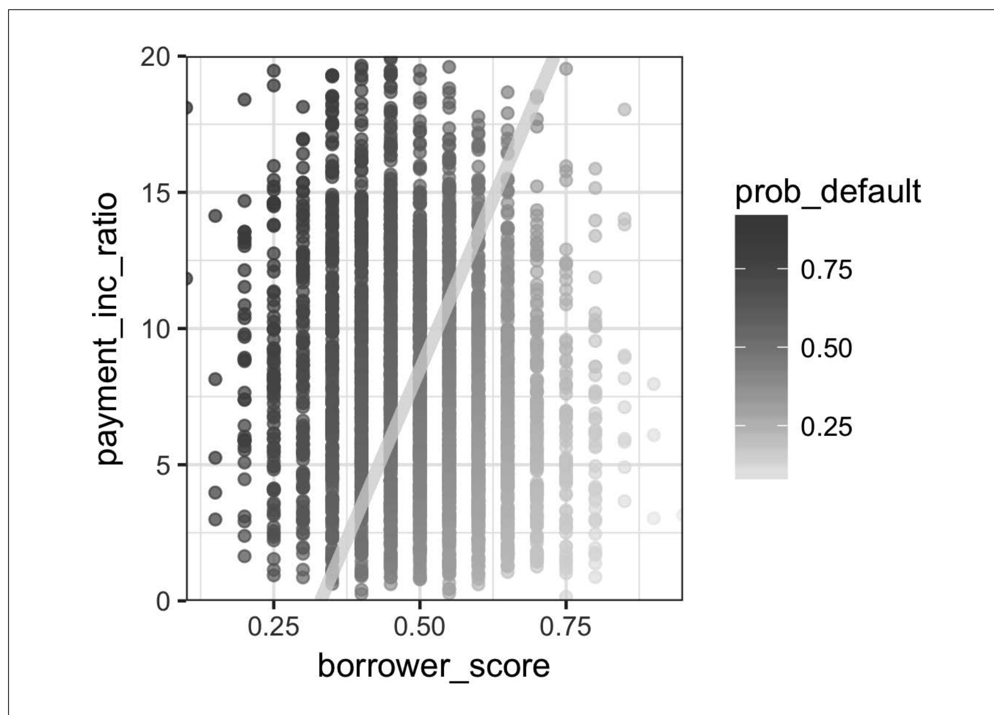


图 5-1：线性判别分析法预测的贷款拖欠。其中使用了两个变量：借款者的信用评分，支出与收入的比率

线性判别分析法预测使用判别函数的权重，将预测空间划分为两个区域，如图中的实线所示。远离实线的预测值，具有更高的置信度（即概率远大于 0.5）。



判别分析的扩展

用于更多的预测变量：尽管本节的内容和例子中仅使用了两个预测变量，但线性判别分析法同样适用于两个以上的预测变量。唯一的限制因素是记录的数量（在估计协方差矩阵时，需要每个变量都具有足够多的记录，但在数据科学应用中，这通常并不是一个问题）。

二次判别分析：判别分析还有其他一些变体，其中最著名的是**二次判别分析**（QDA）。虽然名字这么叫，但二次判别分析依然是一种线性判别函数。两者之间的主要差别在于，对于 $Y = 0$ 和 $Y = 1$ 的情况，线性判别分析假设两组间的协方差矩阵是相同的，而二次判别分析则允许两组间的协方差矩阵是不同的。在实践中，这一差别对于大多数应用并不重要。

本节要点

- 判别分析适用于连续预测因子或分类预测因子，也适用于分类结果。
- 判别分析使用协方差矩阵计算**线性判别函数**，该函数用于区分属于不同类的记录。
- 线性判别函数对每个记录生成一个权重或分值（每个可能的类对应一个权重），以此来确定记录的估计类。

5.2.4　拓展阅读

- Trevor Hastie、Robert Tibshirani 和 Jerome Friedman 合著的《统计学习基础（第 2 版）》，以及 Gareth James、Daniela Witten、Trevor Hastie 和 Robert Tibshirani 合著的《统计学习导论：基于 R 应用》，都有一节的内容介绍了判别分析。
- 在 Galit Shmueli、Peter Bruce 和 Nitin Patel 合著的 *Data Mining for Business Analytics: Concepts, Techniques, and Applications with XLMiner* 一书中，有一整章内容介绍了判别分析。
- 如果对判别分析的历史感兴趣，可以在网上找到费希尔 1936 年发表在 *Annals of Eugenics*（现在更名为 *Annals of Genetics*）上的论文"The Use of Multiple Measures in Taxonomic Problems"。

5.3　逻辑回归

逻辑回归类似于多元线性回归，只是结果是二元的。它使用多种变换将问题转换成可以拟合线性模型的问题。逻辑回归也是一种结构化模型方法，而非以数据为中心的方法。这与判别分析一样，但是不同于 K 最近邻和朴素贝叶斯。逻辑回归的计算速度快，模型输出可以快速地对新数据打分，因此得到了广泛的使用。

主要术语

Logit 函数

一种能将属于某个类的概率映射到 $\pm\infty$ 范围上（而不是 0 到 1 之间）的函数。

同义词：对数几率

几率

"成功"（1）与"不成功"（0）之间的比率。

对数几率

转换后的模型（即线性模型）中的响应。该响应已被映射回概率值。

我们面对的问题是：如何将一个二元结果变量转换为一种可以以线性方式建模的结果变量，然后再转换回二元结果变量？

5.3.1 逻辑响应函数和Logit函数

该问题的关键在于**逻辑响应函数**和 **Logit 函数**。它们实现了将 [0, 1] 区间内的概率值，映射到适用于线性建模的更广的区间上。

首先，我们不能将结果变量简单看作二元标签，而应视为标签是 1 的概率 p。我们可能天真地想将概率 p 建模为预测变量的一个线性函数。

$$p = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q$$

但是在拟合模型时，并不能确保概率 p 位于 [0, 1] 区间内。而作为一个概率值，概率 p 必须位于该区间内。

下面换一种做法。我们通过在预测因子中应用**逻辑响应**函数或**逆 Logit**（inverse logit）函数去建模 p。

$$p = \frac{1}{1 + \mathrm{e}^{-(\beta_0 + \beta_1 x_1 + \beta x_2 x_2 + \cdots + \beta_q x_q)}}$$

这一转换确保了 p 值位于 [0, 1] 区间内。

为了消去分母中的指数表达式，我们考虑使用**几率**（odds）而非概率。对于世界各地的投注者来说，几率无疑是个耳熟能详的词语，它表示“成功”（1）与“不成功”（0）两者间的比率。从概率的角度看，几率是事件发生的概率除以事件不会发生的概率。例如，如果一匹赛马胜出的概率为 0.5，那么“未胜出”的概率就为（1−0.5）= 0.5，这时几率为 1。

$$\mathrm{Odds}(Y = 1) = \frac{p}{1 - p}$$

我们可以使用逆几率函数，由几率得到概率。

$$p = \frac{\mathrm{Odds}}{1 + \mathrm{Odds}}$$

将概率与前面介绍的逻辑响应函数相结合，得到：

$$\mathrm{Odds}(Y = 1) = \mathrm{e}^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q}$$

最后，对等式两端取对数，得到预测因子的一个线性函数表达式。

$$\log(\mathrm{Odds}(Y = 1)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q$$

对数几率（log-odds）函数也称为 **Logit 函数**，它将概率 p 从 [0, 1] 区间映射为 $(-\infty, +\infty)$ 区间上的任何值，如图 5-2 所示。完成这样的转换过程后，我们就可以使用线性模型去预测概率。反过来，我们也可以通过应用**截止规则**（cut-off rules），将概率大于截止值的记录分类为 1，进而将概率值映射为分类值。

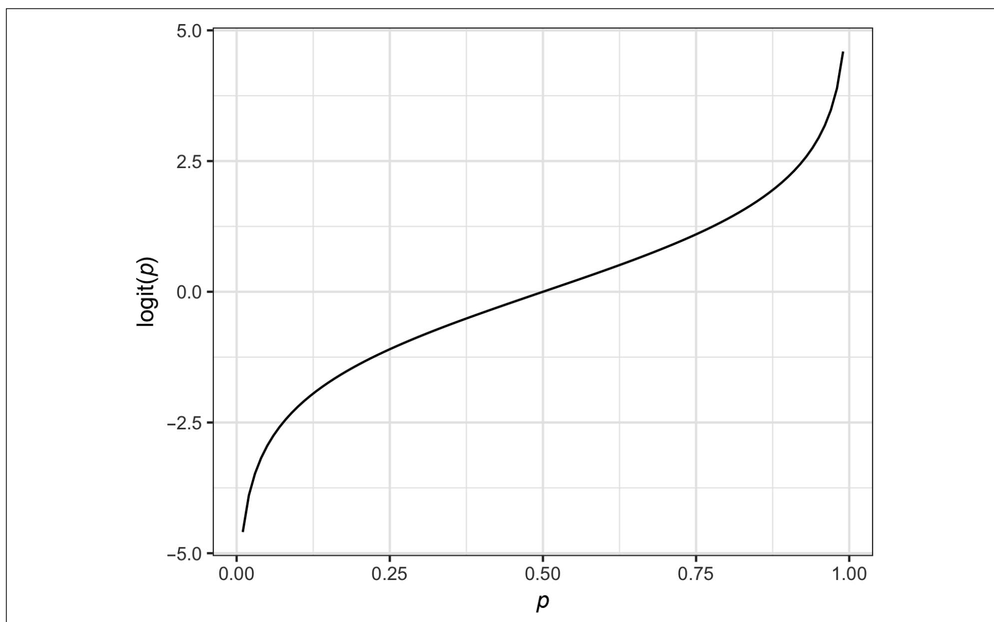


图 5-2：将概率映射到适用于线性模型尺度上的函数（即 Logit 函数）

5.3.2　逻辑回归和广义线性模型

逻辑回归公式中的响应，就是二元结果 1 的对数几率。我们看到的只是二元结果，而不是对数几率，因此需要一种特殊的统计方法去拟合方程。逻辑回归是**广义线性模型**（GLM）的一种特殊实例，用于将线性回归扩展到其他设置。

使用 R 中的 `glm` 函数可以拟合一个逻辑回归，这时需要将函数的参数 `family` 设置为 `binomial`。对于 6.1 节中的个人贷款数据，下面的代码拟合了一个逻辑回归。

```
logistic_model

Call:  glm(formula = outcome ~ payment_inc_ratio + purpose_ + home_ +
    emp_len_ + borrower_score, family = "binomial", data = loan_data)

Coefficients:
               (Intercept)          payment_inc_ratio
                   1.26982                    0.08244
purpose_debt_consolidation    purpose_home_improvement
                   0.25216                    0.34367
    purpose_major_purchase             purpose_medical
                   0.24373                    0.67536
             purpose_other      purpose_small_business
                   0.59268                    1.21226
                  home_OWN                   home_RENT
                   0.03132                    0.16867
         emp_len_ < 1 Year              borrower_score
                   0.44489                   -4.63890
```

```
Degrees of Freedom: 45341 Total (i.e. Null);  45330 Residual
Null Deviance:      62860
Residual Deviance: 57510        AIC: 57540
```

其中，outcome 是响应变量，在还清贷款时值为 0，在欠贷时值为 1。purpose_ 和 home_ 是分别表示贷款目的和房屋产权状态的因子变量。在回归中，具有 P 个层级的因子变量可以用 P–1 个列表示。在 R 中，默认使用了**参考**编码，即将所有的层与参考层做对比（参见 4.4 节）。上述因子的参考层分别是 credit_card 和 MORTGAGE。变量 borrower_score 是一个位于 [0, 1] 区间的分值，表示了借款者的信誉度范围从“差”到“优秀”。该变量是使用 K 最近邻方法从其他几个变量中创建的，具体方法参见 6.1.6 节。

5.3.3　广义线性模型

除了回归之外，另一类重要的模型是**广义线性模型**。广义线性模型的特征主要体现在下面两个方面。

- 一个概率分布或家族。例如，对于逻辑回归，它是二项分布。
- 一个将响应映射到预测因子的**连接函数**（link function）。例如，对于逻辑回归，它是 Logit 函数。

到目前为止，广义线性模型最常见的形式是逻辑回归。数据科学家还会看到其他类型的广义线性模型。有时，我们会发现连接函数使用的是对数函数，而非 Logit 函数。在实践中，对于大多数应用而言，使用对数连接函数基本不会对结果造成很大的差异。泊松分布常用于对计数数据建模，例如用户在一定时间内访问一个网页的次数。其他的分布还包括负二项分布和 Gamma 分布，它们通常用于建模所使用的时间，例如发生故障前正常运行的时间。不同于逻辑回归，使用这些模型的广义线性模型的应用更为微妙，因此在使用中应更为谨慎。除非你熟悉并理解这些方法的使用以及其中的陷阱，否则应尽量避免使用它们。

5.3.4　逻辑回归的预测值

逻辑回归的预测值是以对数几率 $\hat{Y}=\log(\text{Odds}(Y=1))$ 的形式给出的。预测概率可以由逻辑响应函数给出。

$$\hat{p}=\frac{1}{1+\mathrm{e}^{-\hat{Y}}}$$

例如，下面我们查看 logistic_model 模型的预测值。

```
pred <- predict(logistic_model)
summary(pred)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.728000 -0.525100 -0.005235  0.002599  0.513700  3.658000
```

很容易将这些值转换为概率值。

```
prob <- 1/(1 + exp(-pred))
> summary(prob)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.06132 0.37170 0.49870 0.50000 0.62570 0.97490
```

我们可以看到，虽然输出结果位于 [0, 1] 区间内，但是它们并未指明预测值是欠贷还是已偿还贷款。和 K 最近邻分类器一样，我们可将任何大于 0.5 的截止值定义为默认值。在实践中，如果目标是识别罕见类的成员，通常使用较低的截止值（参见 5.4.2 节）。

5.3.5 解释系数和优势比

逻辑回归的一个优点是，其所生成的模型无须重新计算，就可以快速地为新数据打分。另一个优点是，与其他的分类方法相比，其所生成的模型易于解释。对于此，关键理念是对**优势比**的理解。对于二元因子变量 X，优势比最容易理解。

$$优势比 = \frac{\text{Odds}(Y=1 \mid X=1)}{\text{Odds}(Y=1 \mid X=0)}$$

上面的公式可以解释为：当 $X = 1$ 时 $Y = 1$ 的几率与 $X = 0$ 时 $Y = 1$ 的几率的对比。如果优势比为 2，那么表示当 $X = 1$ 时 $Y = 1$ 的几率，是当 $X = 0$ 时 $Y = 1$ 的几率的两倍。

为什么要使用优势比，而不是概率？这是因为逻辑回归中的回归系数 β_j 是 X_j 优势比的对数。

为了更清楚地解释这一问题，下面给出一个例子。回顾 5.3.2 节中拟合的模型，其中 `purpose_small_businessis` 的回归系数为 1.21226。这表示相比于以还清信用卡债务为目的的贷款，贷款给一个小企业可将几率（贷款拖欠对比贷款还清）降低 exp(1.21226)，约等于 3.4。显然，相比于其他用途类型的贷款，以创建或扩大小企业为目的的贷款具有更高的风险。

图 5-3 显示了在优势比大于 1 的情况下，优势比和对数优势比之间的关系。因为回归系数使用了对数尺度，所以回归系数每增加 1，优势比将增加 exp(1)，约等于 2.72。

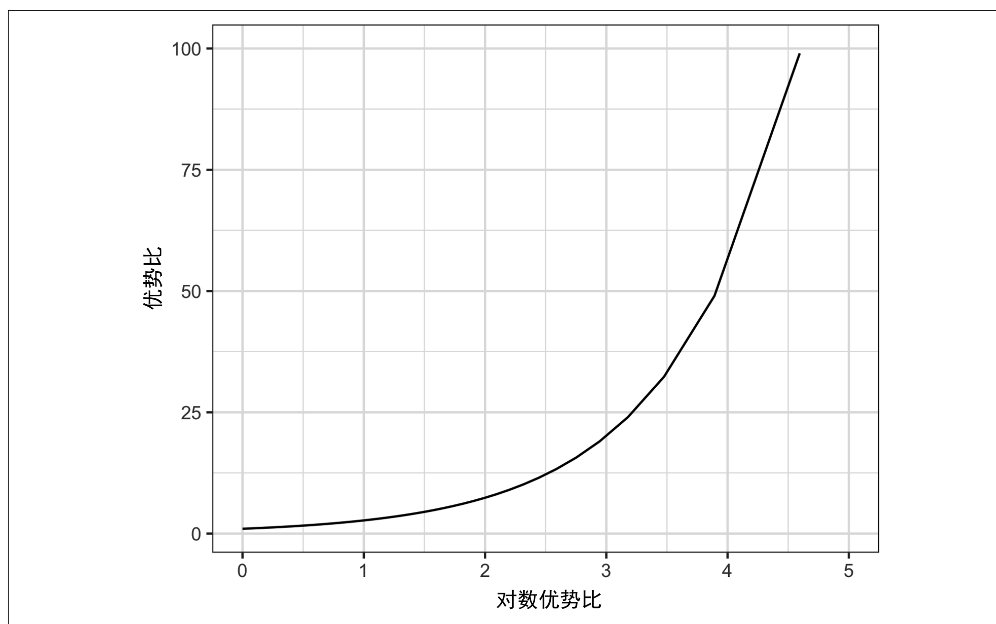


图 5-3：优势比和对数优势比间的关系

对于数值型变量 X 的优势比，也可以做类似的解释。它测量了在 X 发生单位变化时，优势比的变化情况。例如，如果将支出与收入间的比率从 5 增加到 6，那么拖欠贷款的几率将增加 exp(0.08244)，约等于 1.09。变量 `borrow_score` 表示借款者的信用评分，其值的范围从 0（低）到 1（高）。与信用最差的借款者相比，信用最好的借款者的贷款拖欠概率要小 exp(–4.63890)，约等于 0.01。换句话说，信用最差的借款者的贷款拖欠风险是信用最好的借款者的 100 倍！

5.3.6　线性回归与逻辑回归：相似之处和不同之处

多元线性回归和逻辑回归有许多共同点。它们都假设预测因子与响应之间存在线性参数的关联关系，并且都用类似的方式探索并发现最优模型。将模型概化为线性以使用预测因子的样条转换方法，同样适用于逻辑回归。但是，逻辑回归在以下两方面存在根本差异。

- 模型的拟合方式。逻辑回归不适用最小二乘法。
- 模型残差的性质和分析方法。

模型的拟合

线性回归使用最小二乘法拟合，可以使用均方根误差和 R 方等统计量评估拟合的质量。不同于线性回归，逻辑回归并不存在一种封闭的解决方案，模型必须使用**最大似然估计**（MLE）拟合。最大似然估计试图找出一种最有可能生成所见数据的模型。在逻辑回归方程中，响应并非 0 或 1，而是对响应为 1 的对数几率的估计。最大似然估计可以找出一种解决方案，使估计的对数几率能最优地描述所观测到的结果。最大似然估计算法中使用了**拟牛顿优化法**，该机制根据当前的参数值，在打分步骤（即**费希尔分值**）间迭代，逐步更新参数值，以改进模型的拟合度。

> **最大似然估计**
>
> 下面，我们使用统计学符号详细地介绍最大似然估计算法。算法的输入是一组数据 $X_1, X_2, \cdots, X_n$，以及依赖于一组参数 θ 的概率模型 $\mathscr{P}_\theta(X_1, X_2, \cdots, X_n)$。最大似然估计的目标是找出一组参数 $\hat{\theta}$，使得 $\mathscr{P}_\theta(X_1, X_2, \cdots, X_n)$ 的值最大。也就是说，给定模型 $\mathscr{P}_\theta(X_1, X_2, \cdots, X_n)$，最大似然估计最大化观测到 $(X_1, X_2, \cdots, X_n)$ 的概率。在拟合过程中，模型使用**偏差**进行评估。
>
> $$\text{偏差} = -2\log(\mathscr{P}_{\hat{\theta}}(X_1, X_2, \cdots, X_n))$$
>
> 偏差值越低，拟合越好。

幸运的是，大多数用户并不需要关心拟合算法的细节，因为这是由统计软件处理的。大多数数据科学家也不需要关心拟合的方法，只需要知道它能在一定的假设条件下找出好的模型。



处理因子变量

和线性回归一样，我们需要对逻辑回归中的因子变量进行编码，参见 4.4 节。在 R 及其他一些软件中，通常会使用参考编码自动地处理编码问题。本章中介绍的所有其他分类方法一般也使用独热编码表示（参见 6.1.3 节）。

5.3.7　模型评估

和其他分类方法一样，对逻辑回归的评估也依赖于模型对新数据的分类准确程度（参见 5.4 节）。类似于线性回归，我们可以使用一些标准的统计工具去评估并改进模型。除了估计的回归系数，R 还会将回归系数的标准误差、z 值和 p 值等一并给出。

```
summary(logistic_model)

Call:
glm(formula = outcome ~ payment_inc_ratio + purpose_ + home_ +
    emp_len_ + borrower_score, family = "binomial", data = loan_data)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.51951  -1.06908  -0.05853   1.07421   2.15528

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                 1.269822   0.051929  24.453  < 2e-16 ***
payment_inc_ratio           0.082443   0.002485  33.177  < 2e-16 ***
purpose_debt_consolidation  0.252164   0.027409   9.200  < 2e-16 ***
purpose_home_improvement    0.343674   0.045951   7.479 7.48e-14 ***
purpose_major_purchase      0.243728   0.053314   4.572 4.84e-06 ***
purpose_medical             0.675362   0.089803   7.520 5.46e-14 ***
purpose_other               0.592678   0.039109  15.154  < 2e-16 ***
purpose_small_business      1.212264   0.062457  19.410  < 2e-16 ***
home_OWN                    0.031320   0.037479   0.836    0.403
home_RENT                   0.168670   0.021041   8.016 1.09e-15 ***
emp_len_ < 1 Year           0.444892   0.053342   8.340  < 2e-16 ***
borrower_score             -4.638902   0.082433 -56.275  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 64147  on 46271  degrees of freedom
Residual deviance: 58531  on 46260  degrees of freedom
AIC: 58555

Number of Fisher Scoring iterations: 4
```

解释 p 值时的注意事项与回归中一样，我们应将 p 值视为衡量变量重要性的一个相对指标（参见 4.2.2 节），而不是衡量统计显著性的正式标准。具有二元响应的逻辑回归模型没有相关的均方根误差或 R 方。鉴于此，逻辑回归模型通常使用更通用的分类度量评估。具体内容，参见 5.4 节。

线性回归中的许多其他概念，同样可以应用到逻辑回归及其他广义线性模型中。例如，我们可以使用逐步回归、拟合交互项，或是加入样条项。在使用混淆变量和相关变量时需关注的问题，也适用于逻辑回归（参见 4.5 节）。我们可以使用 `mgcv` 软件包拟合广义添加模型（参见 4.7.3 节）。

```
logistic_gam <- gam(outcome ~ s(payment_inc_ratio) + purpose_ +
                        home_ + emp_len_ + s(borrower_score),
                      data=loan_data, family='binomial')
```

逻辑回归的一个不同之处在于对残差的分析。在回归中（参见图 4-9），可以直接计算偏残差。

```
terms <- predict(logistic_gam, type='terms')
partial_resid <- resid(logistic_model) + terms
df <- data.frame(payment_inc_ratio = loan_data[, 'payment_inc_ratio'],
                 terms = terms[, 's(payment_inc_ratio)'],
                 partial_resid = partial_resid[, 's(payment_inc_ratio)'])
ggplot(df, aes(x=payment_inc_ratio, y=partial_resid, solid = FALSE)) +
  geom_point(shape=46, alpha=.4) +
  geom_line(aes(x=payment_inc_ratio, y=terms),
            color='red', alpha=.5, size=1.5) +
  labs(y='Partial Residual')
```

生成的绘图如图 5-4 所示。图中的线条显示了一个估计的拟合，它位于两组点云之间。顶部的点云对应于响应 1（贷款被拖欠），底部的点云对应于响应 0（贷款已还清）。对于输出为二元的逻辑回归，这是非常典型的残差。虽然逻辑回归中的偏残差略逊于回归中残差的意义，但它依然有助于确认非线性行为和识别高影响记录。

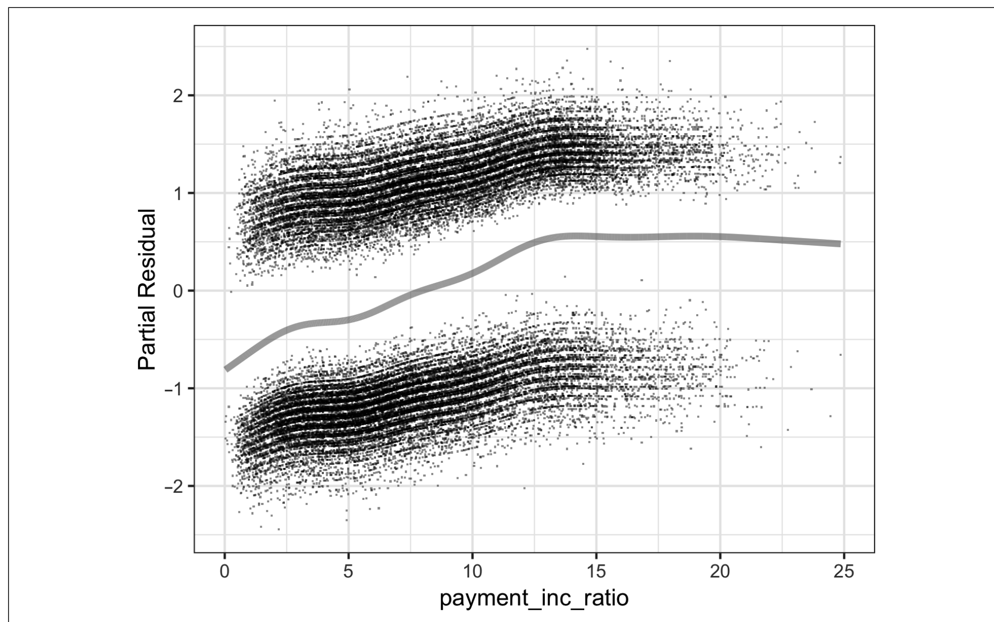


图 5-4：逻辑回归的偏残差



可以忽略 `summary` 函数给出的部分输出。例如，分散度参数并不适用于逻辑回归，它适用于其他类型的广义线性模型。残差偏差和打分计算的迭代次数是与最大似然拟合方法相关的参数。参见 5.3.6 节中的知识点“最大似然估计”。

本节要点

- 逻辑回归和线性回归类似，只不过其结果是二元变量。
- 在逻辑回归中需要做多次转换，以将模型转化为一种可以像线性模型一样拟合的形式，并使用对数优势比作为响应变量。
- 通过迭代过程拟合了线性模型之后，应将对数几率映射回概率值。
- 逻辑回归的计算快速，并且生成的模型可以在不重新计算的情况下对新数据打分，因此它得到了广泛的使用。

5.3.8 拓展阅读

- 逻辑回归的标准参考书，是大 David Hosmer、Stanley Lemeshow 和 Rodney Sturdivant 合著的 *Applied Linear Regression*（3rd ed.）。
- Joseph Hilbe 撰写的两本书也广受欢迎。一本是内容非常全面的 *Logistic Regression Models*，另一本是精炼版的 *Practical Guide to Logistic Regression* 。
- Trevor Hastie、Robert Tibshirani 和 Jerome Friedman 合著的《统计学习基础（第 2 版）》，以及 Gareth James、Daniela Witten、Trevor Hastie 和 Robert Tibshirani 合著的《统计学习导论：基于 R 应用》。两本书都有一节的内容介绍逻辑回归。
- 在 Galit Shmueli、Peter Bruce 和 Nitin Patel 合著的 *Data Mining for Business Analytics* 一书中，有一章专门介绍逻辑回归。

5.4 评估分类模型

人们往往会在预测建模时尝试多个不同的模型，将每个模型应用于一个保留样本（也称**测试样本**或**验证样本**），并评估模型的性能。从本质上看，这就是要查看哪个模型会做出最准确的预测。

主要术语

正确率（accuracy）

正确分类的百分比（或比例）。

混淆矩阵

按预测分类和实际分类情况对记录分别计数，将计数结果以表格形式显示。例如，对于二元变量，使用的是 2×2 的表格。

灵敏度

在预测结果中，1 被正确分类的百分比（或比例）。

同义词：召回率（recall）

特异性

在预测结果中，0 被正确分类的百分比（或比例）。

准确率（precision） 预测结果为 1、真实值也为 1 的百分比（或比例）。 **ROC 曲线** 灵感度与特异性的绘图。 **提升（lift）** 在不同截止概率的情况下，衡量模型在识别（相对罕见的）1 上的有效性。

测量模型分类性能的一种简单方法是，计算预测正确的比例。

在大多数分类算法中，每个预测类都指定了一个“结果为 1 的估计概率”。[3] 默认的决策点或截止值一般采用 0.5 或 50%。如果概率大于 0.5，那么该情况就被分类为“1”；否则，分类为“0”。另一种默认的截止值是使用数据中出现值为 1 的概率。

正确率只是一种对总体误差的度量：

$$\text{正确率} = \frac{(\text{真阳性样本数} + \text{真阴性样本数})}{\text{样本总数}}$$

5.4.1　混淆矩阵

混淆矩阵是分类性能度量的关键所在。混淆矩阵是一种表格，其中显示了按响应类型分类的正确预测数和错误预测数。R 中有多个可以计算混淆矩阵的软件包，但是对于二元变量，采用手动计算更便捷。

下面以 `logistic_gam` 模型为例介绍混淆矩阵。该模型是使用一个平衡数据集训练的，即其中拖欠贷款和还清贷款的数量相同，如图 5-4 所示。遵循惯例，$Y = 1$ 表示感兴趣的事件，在本例中是拖欠贷款；$Y = 0$ 表示负事件（或正常事件），在本例中是贷款还清。对于使用整个训练集（即非平衡数据）训练得到的 `logistic_gam` 模型，下面的 R 代码计算了其混淆矩阵。

```
pred <- predict(logistic_gam, newdata=train_set)
pred_y <- as.numeric(pred > 0)
true_y <- as.numeric(train_set$outcome=='default')
true_pos <- (true_y==1) & (pred_y==1)
true_neg <- (true_y==0) & (pred_y==0)
false_pos <- (true_y==0) & (pred_y==1)
false_neg <- (true_y==1) & (pred_y==0)
conf_mat <- matrix(c(sum(true_pos), sum(false_pos),
                     sum(false_neg), sum(true_neg)), 2, 2)
colnames(conf_mat) <- c('Yhat = 1', 'Yhat = 0')
rownames(conf_mat) <- c('Y = 1', 'Y = 0')
conf_mat
```

注 3：并非每种方法都能给出对概率的无偏估计。在大多数情况下，如果一个方法所给出的排名等价于无偏概率估计所产生的排名，该方法就完全适用。这在功能上等价于使用截止值的方法。

```
        Yhat = 1 Yhat = 0
  Y = 1 14635    8501
  Y = 0 8236     14900
```

在输出中，预测结果按列给出，而真实结果按行给出。我们可以看到，矩阵的对角元素显示了正确预测数，而非对角线元素则显示了错误预测数。在本例中，模型正确地预测了 14 295 个贷款拖欠，但有 8376 个贷款拖欠被错误地预测为已经还清。

图 5-5 显示了二元响应 *Y* 的混淆矩阵，以及混淆矩阵与各种度量之间的关系（更多关于度量的信息，参见 5.4.3 节）。与上面给出的贷款数据的例子一样，实际响应是按行显示的，而预测响应是按列显示的（我们可能也会看到，有一些混淆矩阵的显示与此相反）。对角线上的方格（即左上角和右下角的方格）显示了预测值 $\hat{y}$ 是否正确地预测了响应。其中，**假阳性率**是我们前面并未明确提及的一个重要指标，它是准确率的镜像。当 1 很罕见时，假阳性（FP）与所有预测阳性之间的比率可能会很高，导致无法直观给出预测为 1 但很可能是 0 的情况。一些广泛使用的医学筛查检验，例如乳房 X 光造影，就受到了此问题的困扰。由于病例相对稀少，阳性检测结果很可能并不意味着是乳腺癌。这会导致公众过多的困惑。

真实响应 \ 预测响应	$\hat{y}=1$	$\hat{y}=0$	
$y=1$	True Positive	False Negative	召回率（灵敏度） TP/($y=1$)
$y=1$	False Positive	True Negative	特异性 FP/($y=0$)
发生率 ($y=1$)/总体	准确率 TP/($\hat{y}=1$)		正确率 (TP+TN)/总体

图 5-5：二元响应的混淆矩阵与各种度量。其中，“True Positive”（TP）表示真阳性，“False Positive”（FP）表示假阳性，“False Negative”（FN）表示假阴性，“True Negative”（TN）表示真阴性

5.4.2　稀有类问题

在很多情况下，要预测的类中存在着不平衡的情况，其中一个类比另一个类更普遍，例如，合法保险索赔相对于欺诈保险索赔，浏览购物网站的用户相对于在网站上实际购物的用户。但是，欺诈保险索赔这样的罕见类，往往是我们在分类中更感兴趣的类，一般被指定为 1，以区别于普遍存在的 0。在典型的应用场景中，我们用 1 表示更重要的情况，因为将 1 误分类为 0 要比将 0 误分类为 1 的代价更大。例如，正确识别欺诈保险索赔，会使保险公司免受数千美元的损失。另外，正确识别非欺诈性索赔，你就不必更加仔细地进行手工审核（如果索赔被标为“欺诈性”的，你就会这样做）。

在此类情况下，最准确的分类模型应该将所有的内容分类为 0，除非各个类是非常易于分

离的。例如，如果只有 0.1% 的网店浏览者最终会选择购买，那么预测每个浏览者不购买就离开的模型，其正确率可以达到 99.9%。但是，这样的模型并没什么用处。相反，我们会对一个能从浏览者中挑选出购买者的模型非常满意，尽管该模型可能总体上正确率并不高，会将一些非购买者错误分类。

5.4.3 准确率、召回率和特异性

除了正确率，我们常常还会使用其他一些更精细的度量去评估分类模型。一些度量在统计学尤其是生物统计学中具有悠久的历史，可以描述诊断检验的预期性能。其中，**准确率**测量了预测阳性结果的正确率，如图 5-5 所示。

$$\text{准确率} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FP}}$$

召回率也称为**灵敏度**，它衡量了模型预测阳性结果的能力，即模型正确识别 1 的比例（如图 5-5 所示）。“灵敏度”这一术语多用于生物统计学和医学诊断。而在机器学习领域中，使用更多的是“召回率”。下面是召回率的计算公式。

$$\text{召回率} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FN}}$$

还有一个度量是**特异性**（specificity），它测量了模型预测阴性结果的能力。

$$\text{特异性} = \frac{\sum \text{TN}}{\sum \text{TN} + \sum \text{FN}}$$

```
# 准确率
conf_mat[1,1]/sum(conf_mat[,1])
# 召回率
conf_mat[1,1]/sum(conf_mat[1,])
# 特异性
conf_mat[2,2]/sum(conf_mat[2,])
```

5.4.4 ROC曲线

从上节的定义中可以看出，在召回率和特异性之间存在着权衡。捕获更多的 1，通常意味着有更多的 0 被错误地分类为 1。一个理想的分类器，应该在对 1 的分类上做得很好，不会将更多的 0 分类为 1。

捕获这种权衡的度量，被称为“受试者工作特征”曲线，通常简称为 **ROC 曲线**。ROC 曲线在 y 轴上绘制召回率（灵敏度），在 x 轴上绘制特异性。[4] 当我们更改了分类记录的截止值时，ROC 曲线能显示出召回率和特异性之间的权衡。灵敏度（召回率）绘制在 y 轴上，x 轴的标记可能有两种形式：

注 4：ROC 曲线在第二次世界大战期间被首次使用，用于描述雷达接收站的性能。它的任务是正确地识别（分类）雷达反射信号，并将前来的飞机情况报警给防守部队。

- 特异性绘制在 x 轴上，左边为 1，右边为 0；
- 特异性绘制在 x 轴上，左边为 0，右边为 1。

无论采用哪种方式，ROC 曲线看上去都是相同的。计算 ROC 曲线的步骤如下。

(1) 按 1 的预测概率对记录排序，概率最大的记录在前，概率最小的记录在后。
(2) 根据排序的记录，计算累积特异性和召回率。

在 R 语言中，计算 ROC 曲线很简单。下面的代码计算了贷款数据的 ROC 曲线：

```
idx <- order(-pred)
recall <- cumsum(true_y[idx]==1)/sum(true_y==1)
specificity <- (sum(true_y==0) - cumsum(true_y[idx]==0))/sum(true_y==0)
roc_df <- data.frame(recall = recall, specificity = specificity)
ggplot(roc_df, aes(x=specificity, y=recall)) +
  geom_line(color='blue') +
  scale_x_reverse(expand=c(0, 0)) +
  scale_y_continuous(expand=c(0, 0)) +
  geom_line(data=data.frame(x=(0:100)/100), aes(x=x, y=1-x),
            linetype='dotted', color='red')
```

生成的绘图如图 5-6 所示。图中虚线对角线对应的是一个并不优于随机概率的分类器。对于非常有效的分类器（或医疗中非常有效的诊断测试），ROC 曲线将偏向图的左上角。这样的分类器能够正确地识别大量的 1，不会将很多 0 误分类为 1。对于该模型，如果我们希望分类器具有不低于 50% 的特异性，那么召回率大约为 75%。

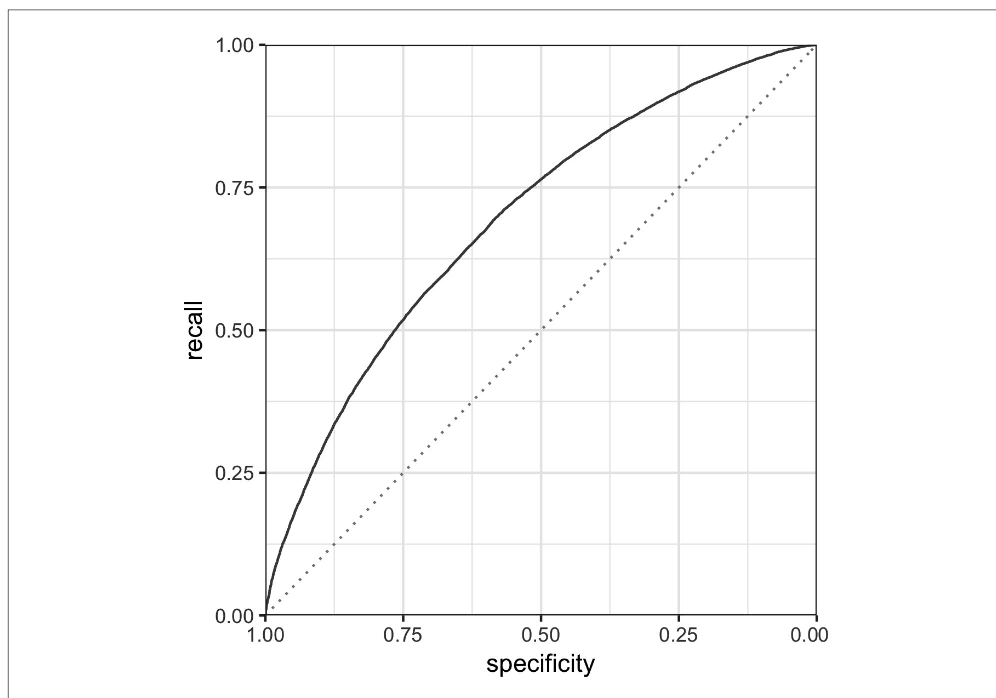


图 5-6：贷款数据的 ROC 曲线



准确率 – 召回率曲线

除了 ROC 曲线之外，准确率 – 召回率（PR）曲线也很有启发意义。PR 曲线的计算方式与 ROC 曲线类似，只是数据按可能性从小到大进行排序，并且计算的统计量是累积准确率和召回率。PR 曲线对于评估具有高度不平衡结果的数据非常有用。

5.4.5 AUC

ROC 曲线是一种十分有用的图形工具，但并非分类器性能的一种度量。然而，我们可以使用 ROC 曲线生成**曲线下面积**（AUC）度量。AUC 就是 ROC 曲线下的总面积。AUC 的值越大，分类器越有效。如果 AUC 为 1，表示一个完美的分类器：将所有 1 正确分类，且没有任何 0 被误分类为 1。

一个完全无效的分类器在 ROC 中显示为对角线，它的 AUC 为 0.5。

图 5-7 显示了贷款数据模型的 ROC 曲线下面积。可以使用数值积分计算 AUC 的值。

```
sum(roc_df$recall[-1] * diff(1-roc_df$specificity))
[1] 0.5924072
```

由此可见，贷款数据模型的 AUC 约为 0.59，对应于一个相对较弱的分类器。

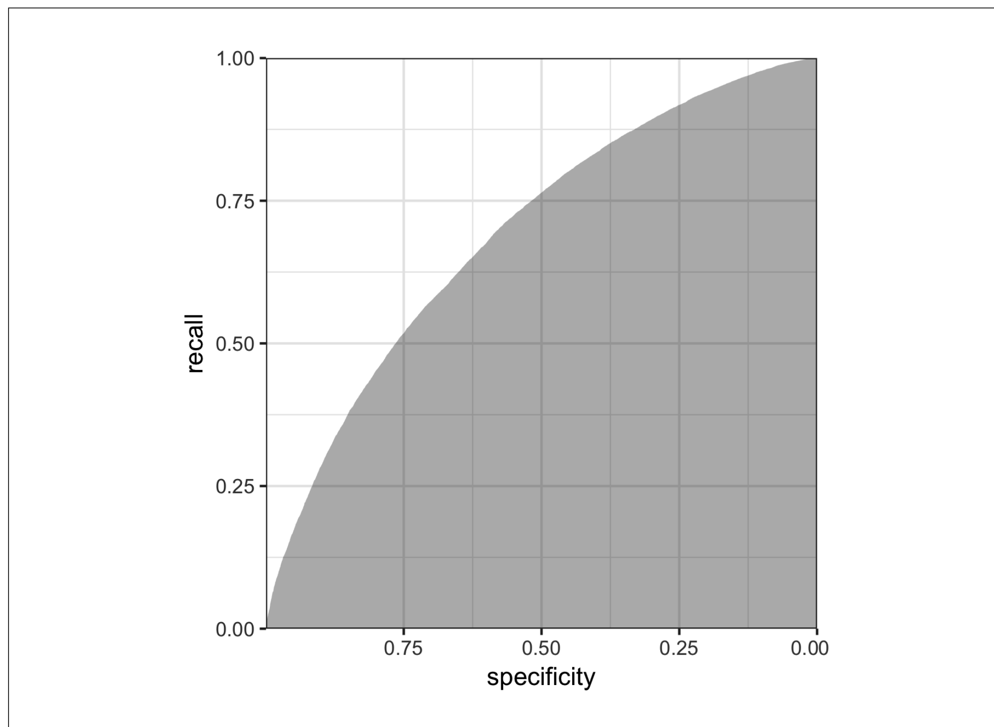


图 5-7：贷款数据的 ROC 曲线下面积



假阳性率的困惑

假阳性率和假阴性率常与特异性或灵敏度相混淆。即便是在一些已发表的著作和软件中，也会出现这样的错误！有时，假阳性率被定义为将真阴性检验为阳性的比例。在很多情况下，例如在网络攻击检测中，该术语用于指阳性信号为真阴性的比例。

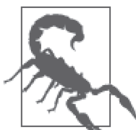
5.4.6 提升

相对于正确率而言，AUC 度量无疑是一种改进，它可以评估一个分类器是否很好地处理了整体正确率与识别更重要的 1 这项需求之间的权衡。但是，AUC 并未完全解决稀有类的问题。在稀有类问题中，为了避免将所有的记录分类为 0，我们需要将模型的截止概率降至 0.5 以下。这时，如果要将一个记录分类为 1，可能需要 0.4、0.3 或更低的截止概率。事实上，我们最终过度识别了 1，这表明这些数据更加重要。

更改截止值会增加捕获 1 的机会，然而代价就是将更多的 0 误分类为 1。那么如何确定最优的截止值？

我们可以使用**提升**（lift）这一概念来解答。我们考虑将记录按预测为 1 的概率进行排序。如果算法使用了分类为 1 的概率来排序前 10% 的记录，那么相比于使用简单盲目选择的基准，算法的效果可以提升多少？如果在排序前 10% 的记录上可以获得 0.3% 的响应，而使用在整体范围内随机挑选的记录可以得到 0.1% 的响应，那么我们称算法在排序前 10% 上的**提升**为 3，也可以称**增益**（gain）为 3。提升图（或增益表）定量地反映了整个数据范围内的提升情况。该图能够以十分位数为单位生成，也可以根据整个数据范围连续生成。

要计算提升图，首先要生成**累积增益图**。在该图中，*y* 轴显示的是召回率，*x* 轴显示的是记录数。**提升曲线**（lift curve）是累积增益与随机选择（对应于对角线）的比率。**十分位数增益图**是预测建模中历史最悠久的技术之一，可以追溯到互联网商务出现之前，尤其受到直邮专业人士的欢迎。如果无差别地使用直邮，那么直邮将是一种费用非常高昂的广告方式。这样，广告商需要使用一种预测模型（在早期是非常简单的模型），识别最有可能产生购买的潜在客户。



抬升（uplift）

有时，我们使用**抬升**表示提升。在某些特定的场景中，抬升还有另一层含义。例如，在做 A/B 测试时，预测模型以处理 A 或处理 B 为预测因子，这时抬升指的是对**单个情况**分别使用处理 A 和使用处理 B 时，预测响应的提高情况。在计算抬升时，首先要将预测因子设为处理 A，对单个情况打分，然后再将预测因子切换成处理 B，再次打分。营销人员和政治竞选顾问使用这种方法来确定应该对客户或选民使用两种处理中的哪一种。

我们可以使用提升曲线查看在不同的概率截止值下将记录分类为 1 的结果。提升曲线可作为设置适当截止值的一个中间步骤。例如，税务机关在做税务审计时，可能资源有限，进而希望能将有限的资源用于发现最可能存在的偷税漏税情况。考虑到资源的限制，税务机

关会使用提升图，绘制所选定的审计报税单和剩余报税单之间的分割线。

本节要点

- 正确率（即预测分类正确的百分比）可以用于评估模型，但只是评估的第一步。
- 其他度量（召回率、特异性、准确率）侧重于更具体的性能特征。例如，召回率测定了模型正确识别 1 的良好程度。
- AUC（ROC 曲线下的面积）是对模型区分 1 与 0 能力的一种常用度量。
- 提升衡量了一个模型在识别 1 上的有效性，并且常常是按十分位数逐个计算的，从分类为 1 可能性最大之处开始。

5.4.7　拓展阅读

评测和评估通常在特定模型的上下文中介绍，例如 *K* 最近邻或决策树。下面三本书用专门的章节介绍了这部分内容。

- Ian Whitten、Elbe Frank 和 Mark Hall 合著的《数据挖掘 : 实用机器学习工具与技术》。
- Benjamin Baumer、Daniel Kaplan 和 Nicholas Horton 合著的 *Modern Data Science with R*。
- Galit Shmueli、Peter Bruce 和 Nitin Patel 合著的 *Data Mining for Business Analytics*。另外，该书有专门适用于 R、Excel 和 JMP 的版本。

下面这本书对交叉验证和重抽样做了很好的介绍：

- Gareth James 等人合著的《统计学习导论：基于 R 应用》。

5.5　不平衡数据的处理策略

上一节介绍了如何使用一些指标（不只是简单的正确率）去评估分类模型。这些指标同样也适用于不平衡数据。在不平衡数据中，我们感兴趣的结果（例如网站上的购买、保险欺诈等）是罕见情况。本节将介绍用于改进不平衡数据的预测建模性能的其他策略。

主要术语

欠采样

在分类模型中，使用更少的多数类记录。

同义词：下采样

过采样

在分类模型中，更多地使用稀有类记录。必要时可以使用自助法。

同义词：上采样

上权重、下权重

在模型中，对稀有类赋予更大的权重，对多数类赋予更小的权重。

数据生成

类似于自助法，只是每个新的自助记录与原记录略有不同。

***z* 分数**

对结果做归一化所生成的值。

K

在最近邻计算中使用的近邻个数。

5.5.1 欠采样

如果有足够多的数据，就像贷款数据的例子中那样，为了使要建模的数据在 0 和 1 之间取得平衡，一种解决方法是对多数类做**欠采样**（或下采样）。欠采样的基本思想是，认为多数类的数据中存在很多冗余的记录。处理规模更小、更平衡的数据集，将有利于改进模型的性能，准备数据以及探索和实验模型也会更容易。

那么多少数据是足够多的？这取决于应用，但一般来说，如果稀有类具有上万条记录，就是足够多的。1 越容易从 0 中区分开来，那么所需的数据就越少。

5.3 节中分析的贷款数据就是基于一个平衡的训练集，其中有一半的贷款被还清，而另一半贷款被拖欠。预测值也给出了类似的结果，有一半预测的概率小于 0.5，另一半预测的概率大于 0.5。但是在完整的数据集中，只有约 18.9% 的贷款被拖欠。

```
mean(loan_data$outcome == "default")
[1] 0.5
mean(full_train_set$outcome=='default')
[1] 0.1889455
```

如果我们使用整个数据集去训练模型，会发生什么情况？

```
full_model <- glm(outcome ~ payment_inc_ratio + purpose_ +
                  home_ + emp_len_+ dti + revol_bal + revol_util,
                  data=train_set, family='binomial')
pred <- predict(full_model)
mean(pred > 0)
[1] 0.003942094
```

可以看到，只有约 0.39% 的贷款被预测为拖欠，或着说，预测结果低于预期值的 1/48。由于模型平等地使用全部数据进行训练，还清贷款的情况压制了拖欠贷款的情况。直观地考虑，由于存在非常多的非拖欠贷款数据，并且预测数据中不可避免地存在变异性，这意味着即便是对于拖欠的贷款，该模型也可能会随机地发现一些非常类似的非拖欠贷款。在使用平衡样本时，约 50% 的贷款被预测为拖欠。

5.5.2 过采样以及上权重和下权重

欠采样中并未使用所有的数据，而是丢弃了一些数据，这是导致欠采样方法饱受批评的一个方面。如果我们面对的是一个规模较小的数据集，其中稀有类包含数百或数千条记录，那么

对多数类做欠采样就会存在丢弃一些有用信息的风险。在这种情况下，我们不应对多数类做欠采样，而应使用有放回的自助法去抽取更多的数据，实现对稀有类的过采样（上采样）。

我们可以通过对数据加权得到类似的效果。很多分类算法采用权重参数，让你可以增减数据的权重。例如，下面的命令使用 glm 函数的 weight 参数，为贷款数据添加了权重向量。

```
wt <- ifelse(full_train_set$outcome=='default',
             1/mean(full_train_set$outcome == 'default'), 1)
full_model <- glm(outcome ~ payment_inc_ratio + purpose_ +
                  home_ + emp_len_+ dti + revol_bal + revol_util,
                  data=full_train_set, weight=wt, family='quasibinomial')
pred <- predict(full_model)
mean(pred > 0)
[1] 0.5767208
```

其中，贷款拖欠的权重设为 $\frac{1}{p}$，其中 p 是贷款拖欠的概率。非拖欠贷款的权重设为 1。拖欠贷款和非拖欠贷款的权重之和大致相等。现在，预测值的均值是 57.7%，不再是 0.39%。

注意，添加权重是对稀有类做过采样和对多数类做欠采样的一种替代方法。



修改损失函数

许多分类和回归算法的优化目标是某一标准或**损失函数**。例如，逻辑回归的目标是尽量最小化偏差。在一些文献中，研究人员建议修改损失函数，以避免发生由稀有类导致的问题。这在实践中是很难做到的，因为分类算法可能会非常复杂，以至于难以对算法本身做出修改。加权是一种更改损失函数的简单方法。使用较低的权重，可以降低记录误差的影响，有利于更高权重的记录。

5.5.3 数据生成

自助法过采样（参见 5.5.2 节）有一种变体，就是通过打乱现有记录来创建新记录的**数据生成**方法。该方法背后的理念是，由于我们只观测到有限组实例，所以算法在构建分类“规则”时并未使用足够丰富的信息。通过创建与现有记录相似但又不完全相同的新记录，可以使算法有机会去学习更强大的规则集。该理念类似于 Boosting 和 Bagging 等集成统计模型的思想（参见第 6 章）。

SMOTE 算法的发表进一步推动了该理念。SMOTE 是 Synthetic Minority Oversampling Technique（合成少数类过采样技术）的缩写形式。该算法找出与过采样记录相似的记录（参见 6.1 节），并对原始记录及其相邻的记录随机加权后取平均，生成一个合成记录，记录中的权重是根据每个预测因子单独生成的。创建的合成过采样记录的数量，取决于使数据集在结果类上取得大致平衡所需的过采样率。

R 语言中提供了多种实现 SMOTE 算法的软件包。其中，unbalance 软件包是最全面的不平衡数据处理软件包。该软件包提供了多种技术，包括一种选择最佳方法的“竞争”算法。其实，SMOTE 算法十分简单，我们可以使用 R 中的 knn 软件包直接实现它。

5.5.4　基于代价的分类

在实践中，正确率和 AUC 是代价最低的分类规则。通常，可以为假阳性与假阴性的比例指定一个估计代价。在确定分类 1 和 0 的最佳截止值中，应该考虑这一代价。例如，假设新贷款拖欠的预期成本是 C，而还清贷款的预期收益是 R。那么贷款的预期收益是：

$$\text{预期收益} = P(Y=0)\times R + P(Y=1)\times C$$

这里，我们并非将贷款简单地标记为“拖欠”或“偿清”，也并非要确定贷款拖欠的概率，更合理的做法是确定贷款是否具有正预期收益。要确定预期的利润情况，预测拖欠概率只是一个中间步骤，它必须与贷款总额一并考虑。而预期的利润才是企业的最终规划度量。例如，相比于预测拖欠概率稍高的大额贷款，较小数额的贷款可能会通过。

5.5.5　探索预测值

AUC 之类的单一指标，并不能捕获在某一情况下一个模型适用性的所有方面。图 5-8 显示了四种不同模型的决策规则。这些模型分别为：线性判别分析、逻辑线性回归、使用广义加性模型拟合的逻辑回归以及树模型（参见 6.2 节），它们都仅使用了 `borrower_score` 和 `payment_inc_ratio` 这两个预测变量去拟合贷款数据。图 5-8 中，线段左上角的区域表示贷款拖欠预测值。我们可以看出，线性判别分析和逻辑线性回归给出了几乎相同的结果，而树模型给出了形状最曲折的规则。事实上，在一些情况下，增加借款者的分值会使预测值从“偿清”转为“拖欠”！最后，使用广义加性模型拟合的逻辑回归是树模型和线性模型这两者的折中。

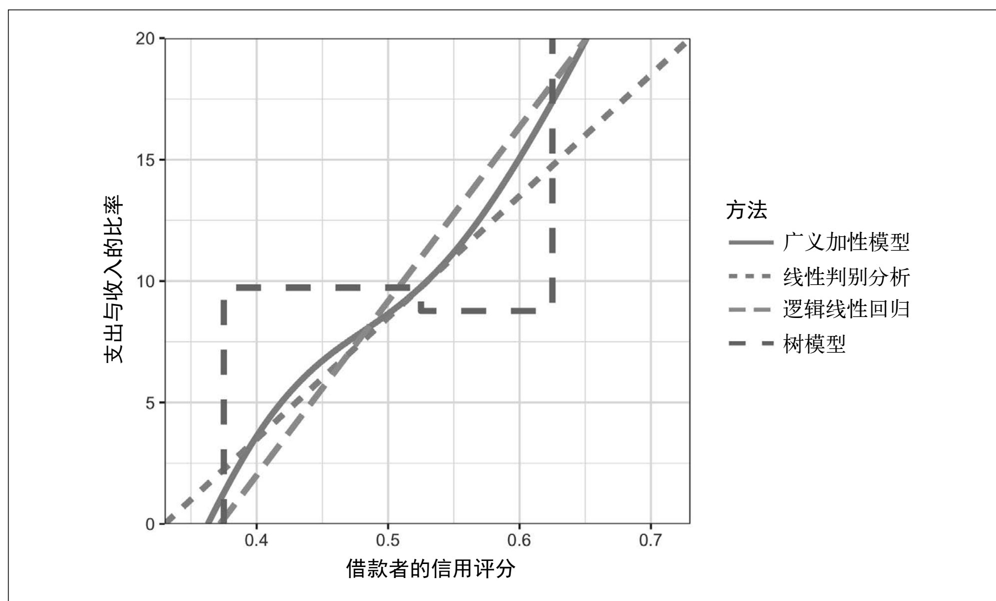


图 5-8：对比四种不同方法的分类规则

在较高的维度上，不容易实现预测规则的可视化。对于广义加性模型和树模型，这些规则的区域也不易于生成。

但无论如何，对预测值做探索性分析终归是有必要的。

> **本节要点**
>
> - 分类算法在高度不平衡数据（其中感兴趣的结果“1”十分罕见）中会存在问题。
> - 平衡训练数据的一种策略是，对多数类做欠采样，或者对稀有类做过采样。
> - 如果使用了数据中所有的“1”依然不够，可以对稀有类做自助法，或使用 SMOTE 算法创建与稀有类相似的合成数据。
> - 不平衡数据通常表明正确的分类（即“1”）具有更高的价值。我们应将这种值的比率纳入到评估度量中。

5.5.6 拓展阅读

- *Data Science for Business* 一书的作者 Tom Fawcett 撰写了一篇介绍不平衡类的优秀文章“Learning from Imbalanced Classes”。
- SMOTE算法的更多细节，参见Nitesh V. Chawla、Kevin W. Bowyer、Lawrence O. Hall和W. Philip Kegelmeyer 合著的文章“SMOTE: Synthetic Minority Over-sampling Technique”。该文发表在 *Journal of Artificial Intelligence Research* 16 (2002): 321–357 上。
- 也可以阅读 Analytics Vidya 网站的内容团队于 2016 年 3 月 28 日发布的文章“Practical Guide to deal with Imbalanced Classification Problems in R”。

5.6 小结

分类是预测记录属于两个（或一组）分类中哪一个的过程，它是预测分析的基础。贷款是否会拖欠？贷款是否会预先还清？ Web 访问者是否会点击一个链接？ Web 访问者是否会网购？一份保险索赔是否存在欺诈？在分类问题中，通常有一个类是我们感兴趣的，如欺诈性保险索赔。在二元分类中，这个类指定为“1”，而更普遍存在的另一个类则指定为“0”。分类过程的关键通常在于估计**倾向性分值**（propensity score），即一个记录属于感兴趣类的概率。一种常见的情景是感兴趣类较为罕见。本章最后介绍了除了简单的正确率之外的多种模型评估度量。这些度量对于稀有类场景非常重要，因为将所有的记录分类为“0”无疑会产生高正确率。

第 6 章

统计机器学习

统计学的最新进展均致力于为回归和分类提供更强大的自动预测建模技术。这些方法都属于**统计机器学习**。不同于经典的统计方法，统计机器学习是数据驱动的，并不试图在数据上强加线性结构或其他的整体结构。例如，*K* 最近邻算法就是一种非常简单的方法，它根据与一个记录相似的记录的分类情况，对该记录进行分类。最为成功并广为使用的技术，是结合**决策树**的**集成学习**。集成学习的基本理念是，运用多个模型而非单一模型去生成预测。决策树对于学习预测变量和结果变量之间关系的规则，是一种灵活且自动化的技术。事实证明，集成学习与决策树相结合，可以得到性能优良的、可用的预测建模技术。

不少统计机器学习技术的发展都可追溯至两位统计学家，他们分别是美国加州大学伯克利分校的利奥 • 布莱曼（Leo Breiman，如图 6-1 所示）和斯坦福大学的杰里 • 弗里德曼（Jerry Friedman）。他们与伯克利和斯坦福大学的其他研究人员一起，在 1984 年开启了**树模型**的发展。随后，在 20 世纪 90 年代，集成学习方法 Bagging 和 Boosting 的发展确立了统计机器学习的基础。



图 6-1：利奥 • 布莱曼，美国加州大学伯克利分校统计学教授，数据科学家所使用的多种核心技术的研究者



机器学习与统计学

机器学习和统计学在预测建模上有哪些差别？二者之间并不存在一条明确的分界线。机器学习更关注如何开发可扩展到大规模数据上的高效算法，以便于优化预测模型。统计学更关注的是概率理论和模型的底层结构。Bagging 算法和随机森林方法（参见 6.3 节）完全是从统计学领域发展出来的。而 Boosting 方法（参见 6.4 节）是从这两个学科中发展起来的，只是在机器学习一方得到了更多的关注。如果不看历史的话，Boosting 的发展确保了该技术同时适用于统计学和机器学习这两个领域。

6.1 *K*最近邻算法

***K* 最近邻**（KNN）算法的理念非常简单。[1] 对于每个要进行分类或预测的记录，该算法：

(1) 找出 *K* 个具有相似特征（即具有相似的预测值）的记录。
(2) 对于分类，找出这些相似记录中的多数类，将其指定为新记录的类。
(3) 对于预测（也称为 **KNN 回归**），找出这些相似记录的均值，并将该均值作为新记录的预测值。

主要术语

近邻
具有相似预测值的两个记录。

距离度量
以单一数值的形式，测量两个记录之间的距离。

标准化
减去均值，并除以标准偏差。

同义词：归一化

***z* 分数**
标准化后得到的值。

K
在最近邻计算中考虑的近邻个数。

K 最近邻算法是一种简单的预测和分类技术，它不像回归那样需要拟合一个模型。但这并不意味着使用 *K* 最近邻算法不需要人工干涉。*K* 最近邻算法的预测结果取决于特征的规模、相似性的测定方法以及 *K* 值的设置等因素。此外，在 *K* 最近邻算法中，所有的预测变量必须是数值型的。下面通过一个分类例子介绍 *K* 最近邻算法。

注 1：本章中各节的内容，版权属于本书作者彼得 · 布鲁斯和安德鲁 · 布鲁斯，© 2017 Datastats，LLC。使用需经许可。

6.1.1 预测贷款拖欠的示例

表 6-1 显示了美国 Lending Club 公司个人贷款数据的部分记录。美国 Lending Club 公司是 P2P 借贷领域的引领者，它将投资者汇集起来，向个人提供贷款。我们数据分析的目标就是预测一笔新贷款的结果是偿清还是拖欠。

表6-1：美国Lending Club公司贷款数据的部分记录

结果	贷款数额	收入	贷款目的	工作时间（单位：年）	住房情况	所在州
偿清	10 000	79 100	债务合并	11	有房贷	内华达州
偿清	9600	48 000	搬家	5	有房贷	田纳西州
偿清	18 800	12 003	债务合并	11	有房贷	马里兰州
拖欠	15 250	23 200	小企业融资	9	有房贷	加利福尼亚州
偿清	17 050	35 000	债务合并	4	租房	马里兰州
偿清	5500	43 000	债务合并	4	租房	堪萨斯州

下面考虑一个非常简单的模型，其中只有两个预测变量：`dti`，表示偿还的债务（不包括房贷）与借款者收入间的比率；`payment_inc_ratio`，表示偿还的贷款与借款者收入的比率。两个比率都乘以 100。我们使用的数据集是有 200 笔贷款的一小组数据，即“loan200”。模型输出是二元预测变量 `loan200`，值为“拖欠”（default）或“偿清”（paid off）。*K* 值设置为 20。下面，我们在 R 中计算在 `dti=22.5` 和 `payment_inc_ratio=9` 的情况下，对要预测的新贷款 `newloan` 的 KNN 估计值。

```
library(FNN)
knn_pred <- knn(train=loan200[-1,2:3], test=newloan, cl=loan200[-1,1], k=20)
knn_pred == 'paid off'
[1] TRUE
```

KNN 的预测值为“偿清”。

虽然 R 语言本身也提供了一个原生的 `knn` 函数，但是由第三方提供的 R 语言软件包 `FNN` 可以更好地扩展到大数据上，并且具有更高的灵活性。

图 6-2 给出了该例子的可视化展示。要预测的新贷款数据位于图的中心处，用小方块标识。圆点（拖欠）和叉号（偿清）标识训练数据。黑色线条所绘制的圆圈显示了 20 个最近邻点的边界。在本例中，圆圈内有 14 个贷款拖欠数据点，而只有 6 个偿清贷款数据点。因此，贷款的预测输出是“拖欠”。



KNN 预测的输出通常是二元决策。例如，在贷款数据中，预测的输出是“拖欠”或“偿清”。KNN 还可以输出一个位于 [0, 1] 区间内的概率值（倾向性）。该概率值基于某个类在 *K* 个最近邻中的比例。在前面的例子中，可以估计贷款拖欠的概率为 14/20，即 0.7。如果使用概率值，那么我们就能使用分类规则，而非简单的多数票（即概率 0.5）。这一点对于不平衡数据尤为重要，参见 5.5 节。例如，如果预测目标是识别一个稀有类的成员，截止值通常设为低于 50%。一种常用方式是将截止值设为稀有事件的概率。

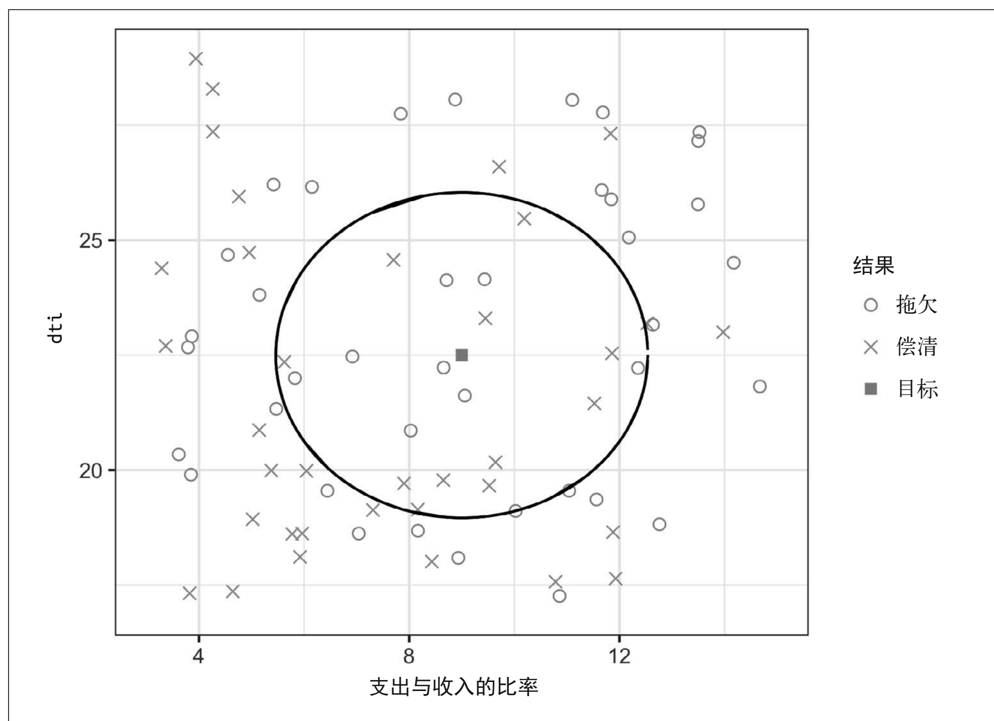


图 6-2：贷款拖欠的 KNN 预测，其中使用了两个变量：贷款与收入的比率，以及偿贷与收入的比率

6.1.2 距离度量

距离度量用于判定相似性（接近度），它是一个测量两个记录 $(x_1, x_2,\cdots, x_p)$ 和 $(u_1, u_2,\cdots, u_p)$ 之间距离的函数。最广为使用的向量距离度量是**欧氏距离**。在测量两个向量间的欧氏距离时，依次取两个向量中对应元素的差值，并对各个差值平方，累加后再取平方根。计算公式为：

$$\sqrt{(x_1-u_1)^2+(x_2-u_2)^2+\cdots+(x_p-u_p)^2}$$

欧氏距离非常容易计算。对于大型数据集，这一点尤为重要，因为 KNN 涉及 $K\times n$ 次逐对比较，其中 n 是向量的行数。

如果数据是数值型的，那么另一种常用的距离度量是**曼哈顿距离**，计算公式为：

$$|x_1-u_1|+|x_2-u_2|+\cdots+|x_p-u_p|$$

欧氏距离表示的是两点之间的直线距离。而曼哈顿距离是在某一时刻以同一方向遍历两点之间的距离，例如遍历矩形城市街区。因此，如果定义相似性为点到点的行程时间，那么曼哈顿距离更适用。

在测量两个向量之间的距离时，测量值取决于规模相对较大的变量（特征）。以贷款数据

为例，距离几乎完全是收入和贷款数额这两个变量的函数，它们是从数万乃至数十万条记录中测量的。相比之下，比率变量几乎不起作用。这个问题可以使用数据的标准化解决，参见 6.1.4 节。



其他距离度量

还有很多测量向量间距离的度量。对于数值型数据，还经常使用**马氏距离**。它考虑了两个变量之间的相关性，这一点十分有用。如果两个变量是高度相关的，那么使用马氏距离就可以在距离上将它们基本视为同一变量。欧氏距离和曼哈顿距离并未考虑相关性，只是对这些特征的属性添加了更大的权重。马氏距离的缺点是在计算中要使用**协方差矩阵**，这增加了计算的难度和复杂性，参见 5.2.1 节。

6.1.3　独热编码

表 6-1 列出的贷款数据中包含了多个因子（字符串）变量。对于大多数统计学和机器学习模型而言，这类变量需要转换为一组承载了同样信息的二元虚拟变量，如表 6-2 所示。我们没有使用一个变量将住房情况表示为“有房产但有房贷”“有房产且无房贷”“租房”或“其他”，而是使用了 4 个二元变量。第一个变量表示“是否有房产但有房贷”，第二个变量表示“是否有房产且无房贷”等。因此，住房情况这个预测变量生成了一个向量，其中包括一个“1”和三个“0”。这样的向量可以用在统计学和机器学习算法中。**独热编码**（one hot encoding）源自数字电路中使用的术语，描述了在电路设置中只允许一个位是正向的（即“热”）。

表6-2：以数值型虚拟变量表示房产情况的因子数据

有房贷	其他	有房	租房
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
0	0	0	1
0	0	0	1



对于线性回归和逻辑回归，独热编码会导致多重共线性问题，参见 4.5.2 节。出现该问题，是因为在编码中忽略了一个虚拟变量，而它的值可以从其他虚拟变量值中推导出来。KNN 和其他方法并不存在这个问题。

6.1.4　标准化

在测量中，我们通常感兴趣的并不是具体的测量结果，而是“与平均值的差异”。**标准化**

（也称为**归一化**）[2] 通过减去均值并除以标准偏差，将所有变量置于同一尺度。该方式避免了变量的原始测量规模对模型产生过度的影响。

$$z = \frac{x - \overline{x}}{s}$$

我们一般称如上标准化的值为 ***z* 分数**。这时，测量值可以用"偏离均值的标准偏差"表示。这样，变量对模型的影响就不会受原始测量规模的影响。



不要将统计学中的**归一化**与**数据库的归一化**混为一谈。数据库标准化的目的在于去除冗余的数据，并验证数据的依赖关系。

在应用 KNN、主成分分析和聚类等算法之前，要考虑数据的标准化问题。为了介绍这一理念，下面对贷款数据应用 KNN 算法（参见 6.1.1 节）。除了使用 `dti` 和 `payment_inc_ratio` 这两个变量之外，我们还添加了另外两个变量：`revol_bal` 表示可提供给借款者的循环信贷总额（单位为美元）；`revol_util` 表示所使用的信贷百分比。下面给出了需要预测的新记录。

```
newloan
  payment_inc_ratio dti revol_bal revol_util
1            2.3932   1      1687        9.4
```

我们看到，变量 `revol_bal`（单位为美元）的规模远大于其他变量。`knn` 函数以属性 `nn.index` 返回了最近邻的索引。下面使用 `nn.index` 显示 `load_df` 中最近的 5 行数据。

```
loan_df <- model.matrix(~ -1 + payment_inc_ratio + dti + revol_bal +
                          revol_util, data=loan_data)
knn_pred <- knn(train=loan_df, test=newloan, cl=outcome, k=5)
loan_df[attr(knn_pred,"nn.index"),]
      payment_inc_ratio  dti revol_bal revol_util
36054           2.22024 0.79      1687        8.4
33233           5.97874 1.03      1692        6.2
28989           5.65339 5.40      1694        7.0
29572           5.00128 1.84      1695        5.1
20962           9.42600 7.14      1683        8.6
```

在显示的 5 个近邻数据中，只有 `revol_bal` 的值与新记录中的对应值非常接近，其他预测变量的值非常分散，在确定近邻上基本没有发挥作用。

下面，我们使用 R 语言的 `scale` 函数对数据做标准化，该函数计算每个变量的 *z* 分数。然后对标准化后的数据应用 KNN。

```
loan_std <- scale(loan_df)
knn_pred <- knn(train=loan_std, test=newloan_std, cl=outcome, k=5)
```

注 2：严格地说，"归一化"（normalization）和"标准化"（standardization）是两种不同的数据规范方法。但是从实现数据按比例缩放这一目的看，可以认为两者实现了相同的目标，尤其是两者都可以采用本节介绍的 *z* 标准化方法实现。归一化的内容，可参见 7.5.1 节。——译者注

```
loan_df[attr(knn_pred,"nn.index"),]
      payment_inc_ratio  dti revol_bal revol_util
2081            2.61091 1.03      1218        9.7
36054           2.22024 0.79      1687        8.4
23655           2.34286 1.12       523       10.7
41327           2.15987 0.69      2115        8.1
39555           2.76891 0.75      2129        9.5
```

可以看到，这时所显示的 5 个近邻数据中，所有的变量都与新记录的对应值非常接近，这样的结果更直观。需要注意的是，虽然结果以是数据的原始比例显示的，但是 KNN 是应用于经过缩放的数据以及要预测的新贷款上的。



z 分数只是一种重新调整变量尺度的方式。均值可以替换成更稳健的位置估计量，例如中位数。同样，标准偏差也可以使用其他的缩放估计量替换，例如四分位距。在一些情况下，变量已被“压缩”到 [0, 1] 区间内。我们也应认识到，为了得到单位变异性而对每个变量进行缩放，其中可能存在一定的随意性。这意味着，我们认为每个变量在预测能力上具有同等的重要性。如果我们只根据自己的主观认识，判定一些变量比其他变量更重要，那么我们可以放大这些变量。例如，对于贷款数据，我们完全有理由认为支出与收入的比率非常重要。



标准化（归一化）并不会改变数据的分布形状。如果数据并不符合正态分布，那么标准化后也不会符合正态分布（参见 2.6 节）。

6.1.5 *K*值的选取

K 值的选取对 KNN 的性能也非常重要。最简单的方法是令 $K = 1$，称为 1- 最近邻分类器。这时预测是直观的，即在训练集中找到与要预测的新记录最相似的数据记录。但是 $K = 1$ 很少是最佳的选择。通常，使用 $K > 1$ 的 KNN 总能获得更好的性能。

一般来说，如果 *K* 值设置过低，可能会产生过拟合，即数据中包含了噪声。而较高的 *K* 值则提供了平滑，降低了过拟合训练数据的风险。另外，如果 *K* 值设置过高，会对数据做过平滑，进而丧失了 KNN 的一个主要优点，即捕获数据局部结构的能力。

通常使用正确率，尤其是测试数据或验证数据的正确率，确定一个能在过拟合和过平滑之间取得最佳平衡的 *K* 值。目前并不存在一种选取最佳 *K* 值的通用规则，这在很大程度上取决于数据的性质。对于具有很少噪声的高度结构化数据，更小的 *K* 值会工作得更好。有时，人们会使用信号处理领域的术语，将这种数据称为具有高**信噪比**（SNR）的数据。高信噪比数据的例子包括手写体识别和语言识别。对于结构松散的噪声数据（即低信噪比数据），例如贷款数据，更大的 *K* 值比较适合。*K* 的常见取值介于 1 ～ 20 之间，通常会选择一个奇数，以免出现平局。



偏差与方差的权衡

过平滑和过拟合之间的拉锯关系，正是**偏差与方差之间的权衡**的一个具体体现。这一问题在统计模型拟合中普遍存在。方差是由于训练数据的选取而产生的建模误差。也就是说，如果选择了不同的训练数据集，那么生成的模型会有所差异。偏差是是由于未能正确识别潜在的现实场景而产生的建模误差。如果只是添加更多的训练数据，偏差并不会消失。当一个灵活的模型产生过拟合时，方差可能会增大。使用一个更简单的模型可以减小方差，但是由于损失了对真实情况建模的灵活性，偏差可能会增大。处理这一权衡的一种通用方法是使用“交叉验证”。具体内容，参见 4.2.3 节。

6.1.6　KNN作为特征引擎

由于其简单性和直观的本质，KNN 得到了广泛应用。但是，与其他更复杂的分类技术相比，KNN 本身在性能上并不具有竞争力。在实际的模型拟合中，可以将 KNN 作为一个阶段性过程，用于向其他分类方法中添加“局部知识”。具体做法如下：

(1) 在数据上运行 KNN，为每个记录生成一个分类（或是分类的拟概率）；

(2) 将结果作为一个新特征添加到记录中，然后在生成的数据上运行另一种分类方法。这样，我们使用了原始预测变量两次。

鉴于部分预测变量在上述过程中使用了两次，你可能会怀疑该过程是否会导致多重共线性（参见 4.5.2 节）。这并不是问题，因为添加到第二阶段模型中的信息只是由数个近邻记录获得的，因而是高度局部的，它们只会构成一种额外信息，而非冗余信息。



我们可以将阶段性地使用 KNN 看作集成学习的一种形式，在集成学习的过程中同时使用了多种预测建模方法。也可以将其视为一种特征工程，目标是提取出一些具有预测能力的特征（预测变量）。在特征工程中，我们通常需要手工审核数据，而 KNN 提供了一种相当自动化的方式。

以美国金县房屋数据为例。在对一个要出售的房屋定价时，房地产经纪人会以**估价**（comps）为基准，即近期售出的类似房屋的价格。事实上，房地产经纪人所做的事情可视为一种手工版的 KNN，即通过查看类似房屋的销售价格对要出售的房屋进行估价。通过将 KNN 应用于近期的销售数据，就可以为统计模型创建一个新特征，以模拟房地产专业人士。我们要预测的值是房屋销售价格，已有的预测变量包括地段、总建筑面积、房屋结构类型、地皮大小，以及卧室数量和浴室数量。由 KNN 生成的新预测变量（特征），就是每条记录的 KNN 预测变量，类似于房地产经纪人的估价。由于我们要预测的是一个数值，因此使用的是 K 个最近邻记录的均值，而不是多数票。这种方法被称为 **KNN 回归**。

同样，对于贷款数据，我们也可以创建一些新特征，表示贷款过程的不同方面。例如，下面的命令将构建一个表示借款者信誉的特征。

```
borrow_df <- model.matrix(~ -1 + dti + revol_bal + revol_util + open_acc +
                          delinq_2yrs_zero + pub_rec_zero, data=loan_data)
```

```
borrow_knn <- knn(borrow_df, test=borrow_df, cl=loan_data[, 'outcome'],
                prob=TRUE, k=10)
prob <- attr(borrow_knn, "prob")
borrow_feature <- ifelse(borrow_knn=='default', prob, 1-prob)
summary(borrow_feature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.4000  0.5000  0.5012  0.6000  1.0000
```

命令的结果是一个特征，表示了基于历史数据对借款者拖欠可能性的预测。

> **本节要点**
>
> - KNN 通过指定与一条记录相似的记录所属的类，实现对该记录的分类。
> - 可以使用欧氏距离或其他相关度量判定相似度（距离）。
> - 与一条记录进行比较的最近邻数（即 K 值），取决于使用不同 K 值时，算法在训练数据上的性能。
> - 预测变量通常需要做标准化，以避免大尺度变量主导了距离度量。
> - KNN 常常作为预测建模过程的第一个阶段。KNN 的预测值会作为一个**预测变量**添加回数据中，进而用于第二阶段（非 KNN）的建模。

6.2 树模型

树模型也被称为**分类与回归树**（CART）[3]、**决策树**（或简称为**树**），它是一种有效的分类和回归方法，最早由利奥·布莱曼等人在 1984 年提出，并得到了广泛的使用。在数据科学中做回归和分类时，树模型及其更加强大的衍生方法**随机森林**和 **Boosting 方法**（参见 6.3 节和 6.4 节），是最广为使用的、也是最强大的预测建模工具。

> **主要术语**
>
> **递归分区（recursive partition）**
>
> 反复对数据进行划分和细分，目的是使每个最终细分内的结果尽可能同质。
>
> **拆分值（split value）**
>
> 一个预测变量值，它将一组记录分为两部分，使得一部分中的预测变量小于拆分值，而另一部分中的预测变量大于拆分值。
>
> **节点**
>
> 在决策树中（或在一组相应的分支规则中），节点是拆分值的图形化表示（或规则表示）。
>
> **叶子**
>
> 一组 `if-then` 规则的终点，或一个树分支的终点。在树中访问叶子的规则，构成了对树中一条记录的分类规则。

注 3：CART 是美国 Salford Systems 公司的注册商标，特指该公司的树模型实现。

损失

在拆分过程的某一阶段中误分类的个数。损失越大，不纯度越高。

不纯度

表示在数据的一个细分中发现多个类混杂的程度。细分中混杂的类越多，该细分的不纯度就越高。

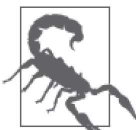
同义词：异质性

反义词：同质性、纯度

剪枝

为了降低过拟合，对一棵完全长成树逐步剪枝的过程。

树模型由一组 `if-then-else` 规则构成，易于理解，也易于实现。树模型不同于回归和逻辑回归，它可以发现数据中隐含的一些复杂交互模式。简单树模型也不同于 KNN 和朴素贝叶斯，它可以表示为预测变量之间的关系，易于解释。



运筹学中的决策树

在决策科学和运筹学中，**决策树**一词有着不同（且更古老）的意义，它指的是人类决策的分析过程。在这种意义下，分支图中绘制的是决策点、可能的结果以及它们的估计概率，并选择具有最大期望值的决策路径。

6.2.1　一个简单的例子

在 R 语言中，拟合树模型主要使用软件包 `rpart` 和 `tree`。下面的代码使用 `rpart` 软件包，根据变量 `payment_inc_ratio` 和 `borrower_score` 对贷款数据拟合了一个树模型。所使用的数据是一个具有 3000 条记录的样本（参见 6.1 节）。

```
library(rpart)
loan_tree <- rpart(outcome ~ borrower_score + payment_inc_ratio,
                   data=loan_data, control = rpart.control(cp=.005))
plot(loan_tree, uniform=TRUE, margin=.05)
text(loan_tree)
```

树模型如图 6-3 所示。图中给出的分类规则是通过遍历层级树模型确定的。遍历从树的根部开始，直到抵达了一个叶子。

绘制的树通常是倒置的，即根显示在顶部，叶子显示在底部。如果贷款变量 `borrower_score` 的值为 0.6，变量 `payment_inc_ratio` 的值是 8.0，结果最终会落在最左边的叶子上，即预测贷款将会被还清。

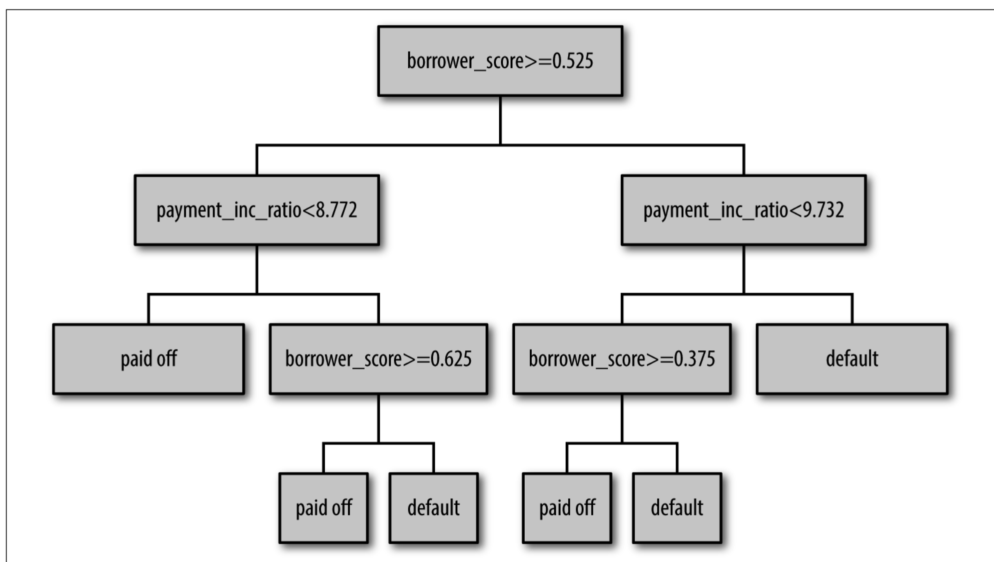


图 6-3：拟合贷款数据的一个简单树模型中的规则

使用 R 语言，很容易生成一个绘图效果更好的树模型。

```
loan_tree
n=3000

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 3000 1467 paid off (0.5110000 0.4890000)
  2) borrower_score>=0.525 1283  474 paid off (0.6305534 0.3694466)
    4) payment_inc_ratio< 8.772305 845  249 paid off (0.7053254 0.2946746) *
    5) payment_inc_ratio>=8.772305 438  213 default (0.4863014 0.5136986)
     10) borrower_score>=0.625 149   60 paid off (0.5973154 0.4026846) *
     11) borrower_score< 0.625 289  124 default (0.4290657 0.5709343) *
  3) borrower_score< 0.525 1717  724 default (0.4216657 0.5783343)
    6) payment_inc_ratio< 9.73236 1082  517 default (0.4778189 0.5221811)
     12) borrower_score>=0.375 784  384 paid off (0.5102041 0.4897959) *
     13) borrower_score< 0.375 298  117 default (0.3926174 0.6073826) *
    7) payment_inc_ratio>=9.73236 635  207 default (0.3259843 0.6740157) *
```

在输出结果中，以缩进的形式表示了树的深度。树的每个节点对应一个临时分类，该临时分类由该分区中的主要结果确定。输出中的 `loss` 项表示每个分区的临时分类所产生的误分类数。例如，在节点 2，在 1467 条记录中有 474 条被误分类。括号中的值分别对应已偿清和拖欠记录的比例。例如，在预测为拖欠的节点 13 中，有 60% 以上的记录是贷款拖欠。

6.2.2　递归分区算法

决策树的构造算法被称为**递归分区法**，该算法的理念十分直观，运算也非常直接。它通过选取预测变量值，将分区中的数据划分为同质的子分区，进而重复地对当前分区数据

进行拆分。图 6-4 显示了根据图 6-3 的树模型所创建的分区。第一条规则是 `borrower_score>=0.525`，图中表示为“1”。第二条规则是 `payment_inc_ratio<8.772`，它将“1”划分出的右侧分区再次拆分为两个子分区。

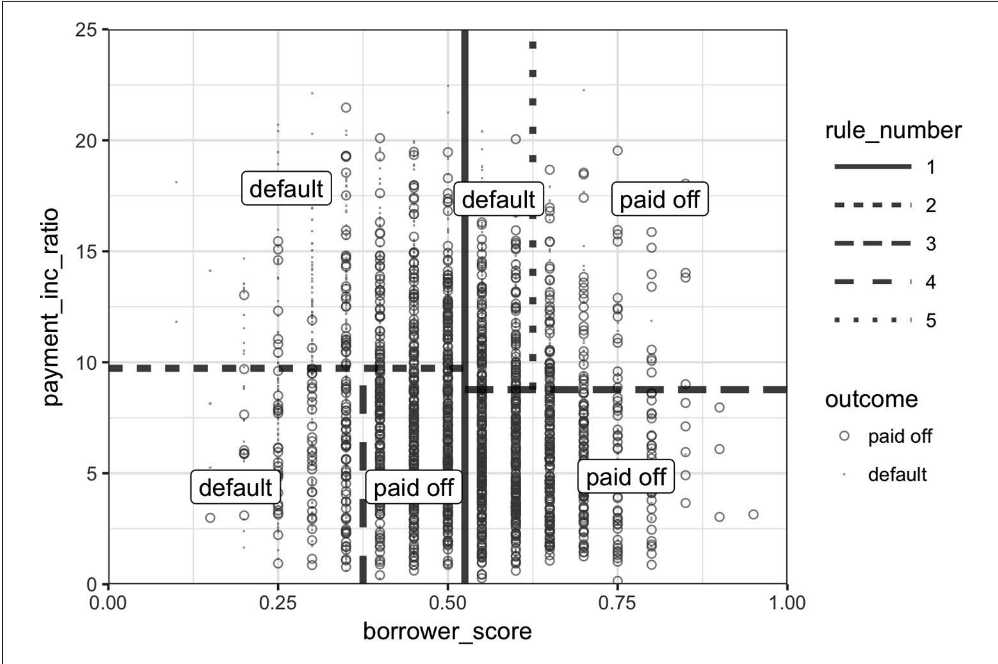


图 6-4：拟合贷款数据的一个简单树模型中的规则

假设我们有一个响应变量 Y 和一组 P 个预测变量 X_j（$j = 1,\cdots, P$）。对于分区 A 中的记录，递归分区算法将找出将 A 拆分为两个子分区的最佳方法：

(1) 对于每个预测变量 X_j，

　a. 对于 X_j 中的每个值 s_j：

　　– 将 A 中的记录拆分为 $X_j<s_j$ 和 $X_j \geqslant s_j$ 两个子分区

　　– 对 A 的每个子分区，测量其中类的同质性

　b. 选择生成了分区内最大类同质性的 s_j。

(2) 选择生成了分区内最大类同质性的拆分值 s_j 和变量 X_j。

下面给出算法中的递归计算部分。

(1) 初始化，以分区 A 为整个数据集。
(2) 应用上面的分区算法，将分区 A 拆分为两个子分区 A_1 和 A_2。
(3) 对子分区 A_1 和 A_2 重复步骤 2。
(4) 如果进一步分区不再从实质上改进子分区的同质性，终止算法。

最终结果就是类似于图 6-4 所示的数据分区，一个不同之处是数据为 P 维的，另一个不同之处是每个分区会根据该分区中响应变量的多数票情况，给出一个值为“0”或“1”的预测。



树模型不仅可以做出值为“0”或“1”的二元预测，还可以根据分区中的“0”和“1”的数量产生一个概率估计值。概率估计值是分区中“0”或“1”的总个数，除以分区中观测值的总个数。

概率 $(Y = 1)$ = 分区中“1”的个数 / 分区大小

$Y = 1$ 的概率估计值可以转换为二元决策。例如，如果概率估计值大于 0.5，那么可以设估计值为 1。

6.2.3 测量同质性或不纯度

树模型递归地创建分区（记录的集合），并给出 $Y = 0$ 或 $Y = 1$ 的预测结果。从上节介绍的递归算法中可以看到，我们还需要一种测量分区同质性（也称为**类纯度**）的方法。或者也可以说，我们需要测量分区的不纯度。预测的正确率是分区内误分类记录的比例 p。p 的取值介于 0（即完美分区）和 0.5（即纯随机猜测）之间。

事实证明，对于不纯度来说，正确率并非一种很好的度量。两种常用的不纯度度量分别是**基尼不纯度**和**熵**（或**信息**）。虽然这些不纯度度量及其他一些度量适用于多分类（即两个以上的类）问题，在此我们只介绍二元的情况。一组记录 A 的基尼不纯度 $I(A)$ 定义为：

$$I(A) = p(1 - p)$$

熵 $I(A)$ 由下式给出：

$$I(A) = -p \log_2(p) - (1 - p)\log_2(1 - p)$$

图 6-5 显示了基尼不纯度（经过重新缩放的）与熵度量是相似的，在中等正确率和高正确率的情况下，熵会给出较高的不纯度分值。

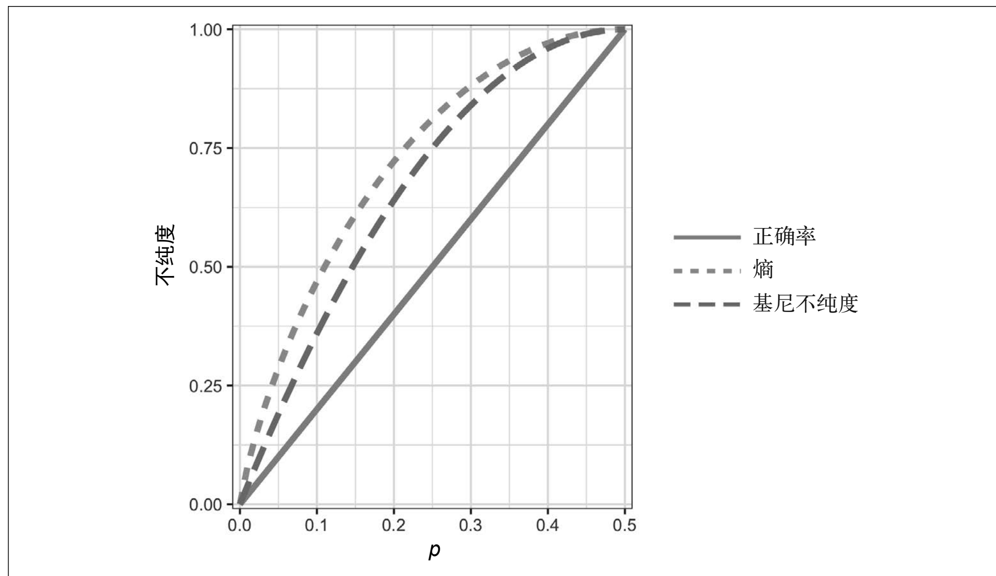
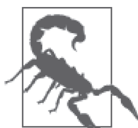


图 6-5：基尼不纯度和熵度量



基尼系数

不要将基尼不纯度与**基尼系数**混淆。虽然它们表示相似的概念，但是基尼系数仅限于二分类问题，并与 AUC 度量有关（参见 5.4.5 节）。

在上一节介绍的递归分区算法中，我们使用了不纯度作为度量。对于该算法给出的每个数据分区，算法测量由拆分所产生的每个子分区的不纯度，然后计算加权均值，并选择（在每个阶段中）生成最低加权均值的子分区。

6.2.4　阻止树模型继续生长

随着树变得越来越大，拆分规则变得越来越详细，树也逐渐从识别确定数据中真实可靠关系的“大”规则，转变为识别仅反映噪声的“小”规则。一棵完全长成树会生成完全纯粹的叶子，进而在训练数据上达到 100% 的分类正确率。

当然，这个正确率是不实际的——树过拟合了数据（参见 6.1.5 节中的知识点“偏差与方差的权衡”）。这表明我们在训练数据中拟合了噪声，而不是我们想要在新数据中识别的信号。



剪枝

一种减小树规模的简单又直观的方法，是对树的末端和较小的分支做**剪枝**，从而生成较小规模的树。那么我们要对树剪枝到哪一层？一种常用的技术是剪枝到验证数据集上的误差达到最小。如果组合了来自多个树模型的预测（参见 6.3 节），就需要一种能阻止树模型生长的方法。在使用交叉验证确定树在组合方法中的生长程度时，剪枝会发挥作用。

对于某个阶段，我们需要一种方法确定何时停止树模型的生长。该方法也可泛化到新数据。常用的停止拆分方法有以下两种。

- 如果拆分后的子分区过小，或末端叶子的规模过小，就应避免拆分。在 `rpart` 软件包中，这些约束是由参数 `minsplit` 和 `minbucket` 控制的，它们的默认值分别是 20 和 7。
- 如果新分区并未使不纯度“显著”降低，那么就不必拆分该分区。在 `rpart` 软件包中，这是由**复杂度参数** `cp` 控制的。该参数度量了树模型的复杂度。树模型越复杂，`cp` 的值就越大。在实践中，参数 `cp` 是通过对树模型的额外复杂度（拆分）附加惩罚项，从而限制了树模型的增长。

在第一种方法中，我们可以添加任意规则，因此它适用于探索性工作。但是该方法不易确定最优值，即让新数据的预测准确性最大化的值。使用复杂度参数 `cp`，可以确定在新数据上性能最好的树模型规模。

如果 `cp` 值过小，那么树模型将过拟合数据，即模型拟合了噪声而非信号。反过来，如果 `cp` 值过大，那么树模型的规模将过小，预测能力会很低。参数 `rpart` 的默认值为 0.01，但对于较大的数据集，我们或许会发现该默认值还是偏大。在前面的例子中，`cp` 值设为 0.005，因为使用默认值会生成一个只有一次拆分的树模型。在探索性分析中，只需对该值做一些简单的尝试即可。

如何确定最优 `cp` 值，这是偏差与方差间权衡问题的又一个实例（参见 6.1.5 节中的知识点“偏差与方差的权衡”）。最常用的 `cp` 值估计方法是使用交叉验证（参见 4.2.3 节），步骤如下。

(1) 将数据分为训练集和验证集。
(2) 使用训练集生长树模型。
(3) 逐步剪枝，并在每步记录 `cp` 值（使用**训练集**）。
(4) 注意在**验证集**上取得最小误差（损失）的 `cp` 值。
(5) 将数据重新拆分为训练集和验证集，并重复树模型的生长、剪枝和记录 `cp` 过程。
(6) 重复执行上述步骤，对反映每个树的最小误差的 `cp` 值求平均值。
(7) 回到原始数据，也可以是将要处理的数据上，生长树模型，并在最优 `cp` 值处终止执行算法。

在 `rpart` 软件包中，可以使用参数 `cptable` 生成一个表，记录 `cp` 值及相关联的交叉验证误差（即 R 语言中的 `xerror`），进而确定具有最低交叉验证误差的 `cp` 值。

6.2.5　预测连续值

使用树模型预测连续值（也被称为**回归**）时，遵循着同样的逻辑和过程，不同之处是使用每个子分区中距离均值的平方偏差（平方误差）来度量不纯度，并通过每个分区中的均方误差的平方根判断预测性能（参见 4.2.2 节）。

6.2.6　如何使用树模型

企业中的预测建模者面临的一大障碍，就是所使用的预测方法在本质上是一种“黑箱”，而“黑箱”操作会引发企业中其他人员的反对。针对这一问题，树模型至少在以下两个方面颇具吸引力。

- 树模型提供了一种可视化的数据探索工具，有助于人们掌握重要的变量，以及这些变量之间是如何关联的。树模型可以捕获预测变量之间的非线性关系。
- 树模型所提供的一套规则可以有效地与非专业人士交流。这有助于数据挖掘项目的实施或“推销”。

然而，对于预测而言，从多个树模型中获取的结果通常要比使用单个树更为强大。特别是随机森林和 Boosting 算法几乎总能提供优越的预测准确性和性能（参见 6.3 节和 6.4 节），但也失去了前述单个树模型的优点。

本节要点

- 决策树生成一组规则，用于分类或预测结果。
- 规则对应于如何将数据划分为连续的子分区。
- 每个分区或拆分指定一个预测变量值（即拆分值），将分区中数据拆分为高于和低于该拆分值的两组记录（即子分区）。
- 在每个阶段，树算法选择使每个子分区内结果的不纯度最小的拆分。

- 一旦算法不能做进一步的拆分，就得到了一棵完全长成树。每个末端节点或叶子内的记录属于相同的类。此后，遵循该规则（拆分）路径的新记录，将会分配为该类。
- 完全长成树会过产生拟合，因此为了使模型捕获信号而非噪声，必须做剪枝。
- 虽然随机森林和 Boosting 等多树模型算法具有更好的预测性能，但失去了单个树模型基于规则的交流能力。

6.2.7 拓展阅读

- Analytics Vidhya 网站的内容团队于 2016 年 4 月 12 日发表的博客文章“A Complete Tutorial on Tree Based Modeling from Scratch (in R & Python)”。
- Terry M. Therneau、Elizabeth J. Atkinson 和梅奥基金会（Mayo Foundation）于 2015 年 6 月 29 日发表的技术报告“An Introduction to Recursive Partitioning Using the RPART Routines”。

6.3 Bagging和随机森林

1907 年，统计学家弗朗西斯·加尔顿爵士（Sir Francis Galton）造访了英国一个县的集市。在集市上举行了一场比赛，竞猜一头参展公牛屠宰后的重量。当时有 800 人给出了自己的猜测，尽管各个猜测值千差万别，但是这些猜测值的均值和中位数均落在真实重量上下浮动 1% 的范围内。James Suroweicki 在其所著的《百万大决定：世界是如何运作的？》一书中探讨了该现象。该原则同样适用于预测模型：如果组合使用多种模型，即对多个模型取平均或多数票，结果要比仅选用一个模型更准确。

主要术语

集成

使用一组模型给出预测。

同义词：模型平均

Bagging

对数据使用自助法构建一组模型的通用方法。

同义词：自助法聚合

随机森林

使用决策树的一类自助法聚合估计。

同义词：自助法聚合决策树

变量重要性

对预测变量在模型性能中重要性的测量。

集成方法已应用于多种不同的建模方法，并在 Netflix 竞赛中得到了使用。只要参赛者提出的模型能将 Netflix 客户的电影评分预测提升 10%，就可获得 Netflix 提供的 100 万美元的

奖金。集成方法的基本实现如下。

(1) 给定一个数据集，采用一种预测模型，并记录该模型的预测情况。
(2) 在同一数据集上，依次使用多个模型重复步骤 1。
(3) 对于每个要预测的记录，对预测值取均值（或加权均值，也可以使用多数票）。

集成方法在决策树中得到了最系统、最有效的应用。集成树模型是一种强大的建模方法，易于构建出好的预测模型。

除了上面的简单集成算法，集成模型还有两种主要的变体：Bagging 和 Boosting。使用了树模型的集成方法被称为**随机森林**模型，或者**提升决策树**（boosted tree）模型。本节内容主要侧重于 Bagging，而 Boosting 将在 6.4 节中介绍。

6.3.1　Bagging方法

Bagging 方法最早是由利奥·布莱曼在 1994 年提出的，Bagging 是 bootstrap aggregating（自助法聚合）的缩写。

Bagging 方法和集成的基本算法类似，只是 Bagging 并不是要在同一数据上拟合所有的模型，而是对使用每个自助法重抽样拟合一个新模型。给定响应变量 Y 和具有 P 个预测变量 $X = X_1, X_2, \cdots, X_p$ 的 n 条记录，下面给出形式化的算法描述。

(1) 初始化要拟合的模型数 M 和要选取的记录数 n（$n < N$）。设置迭代计数 $m = 1$。
(2) 对训练数据做一次具有 n 条记录的自助重抽样（即有放回的抽样），得到子样本 Y_m 和 X_m（即“bag”）。
(3) 使用 Y_m 和 X_m 训练模型，创建一组决策规则 $\hat{f}_m(X)$。
(4) 增加模型计数器，$m = m + 1$。如果 m 不大于 M，则返回第 1 步。

一旦 $\hat{f}_m$ 预测了 $Y = 1$ 的概率，那么自助法聚合估计可由下式给出：

$$\hat{f} = \frac{1}{M}(\hat{f}_1(X) + \hat{f}_2(X) + \cdots + \hat{f}_M(X))$$

6.3.2　随机森林

随机森林是将 Bagging 方法应用于决策树，并做了一个重要的扩展。该算法不仅对记录做抽样，而且也对变量做抽样[4]。传统的决策树在确定如何将一个分区 A 拆分为子分区时，通过最小化基尼不纯度（参见 6.2.3 节）等标准去选择变量和拆分点。在随机森林算法中，每一阶段的变量选择受限于变量的一个**随机子集**。与基本的树算法（参见 6.2.2 节）相比，随机森林算法额外添加了两步，分别是在 6.3.1 节中介绍的 Bagging 方法，以及每次拆分时对变量的自助法抽样。随机森林算法的步骤如下。

(1) 从**记录**中做一次自助法（带放回的）抽样，得到一个子样本。

注 4：注意，**随机森林**是一个注册商标。商标归属于利奥·布莱曼和阿黛尔·卡特勒，并授权给 Salford Systems 公司。“随机森林”一词并不存在标准的非注册商标名称，它几乎等同于该算法，正如“Kleenex”一词几乎等同于面巾纸一样。

(2) 对于第一次拆分，无放回地随机抽样 p（$p<P$）个**变量**。

(3) 对于每组抽样变量 $X_{j(1)}, X_{j(2)}, \cdots, X_{j(p)}$，应用如下的拆分算法。

  a. 对于 $X_{j(k)}$ 的每个值 $s_{j(k)}$：

   – 将分区 A 中满足 $X_{j(k)}<s_{j(k)}$ 的记录拆分为一个分区，其余满足 $X_{j(k)} \geqslant s_{j(k)}$ 的记录作为另一个分区；

   – 测量 A 的每个子分区中类的同质性。

  b. 选择生成分区内最大类同质性的 $s_{j(k)}$。

(4) 选择生成分区内最大类同质性的变量 $X_{j(k)}$ 和拆分值 $s_{j(k)}$。

(5) 继续下一次拆分，重复从步骤 2 开始的上述步骤。

(6) 遵循同一过程，继续拆分，直到得到一棵完全长成树。

(7) 返回步骤 1，再做一次自助法抽样，得到子样本，并重复上述过程。

那么每一步需要抽样的变量数是多少呢？一条经验规则是选取 $\sqrt{P}$ 个，其中 P 是预测变量的个数。在 R 语言中，软件包 `randomForest` 提供了一种随机森林实现。下面的代码对贷款数据应用该软件包（对于贷款数据的介绍，参见 6.1 节）。

```
> library(randomForest)
> rf <- randomForest(outcome ~ borrower_score + payment_inc_ratio,
                     data=loan3000)
Call:
 randomForest(formula = outcome ~ borrower_score + payment_inc_ratio,
              data = loan3000)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 38.53%
Confusion matrix:
         paid off default class.error
paid off     1089     425   0.2807133
default       731     755   0.4919246
```

默认情况下，软件将训练 500 个树模型。鉴于在本例的预测集中只有两个变量，算法将在每一步随机选取是拆分的变量，即自助法抽样得到的子样本的规模为 1。

误差的**包外**（OOB）估计是指将训练得到的模型作用于训练集中未使用的数据上时，所得到的错误率。使用模型的输出，可以绘制出 OOB 误差与随机森林中树模型的数量。

```
error_df = data.frame(error_rate = rf$err.rate[,'OOB'],
                      num_trees = 1:rf$ntree)
ggplot(error_df, aes(x=num_trees, y=error_rate)) +
  geom_line()
```

结果如图 6-6 所示。从图中可以看到，错误率在超过 0.44 处迅速降低，并在约 0.385 处稳定下来。我们可以使用如下代码，从 `predict` 函数得到预测值并绘图。

```
pred <- predict(loan_lda)
rf_df <- cbind(loan3000, pred_default=pred[,'default']>.5)
ggplot(data=rf_df, aes(x=borrower_score, y=payment_inc_ratio,
                       color=pred_default, shape=pred_default)) +
  geom_point(alpha=.6, size=2) +
  scale_shape_manual( values=c( 46, 4))
```

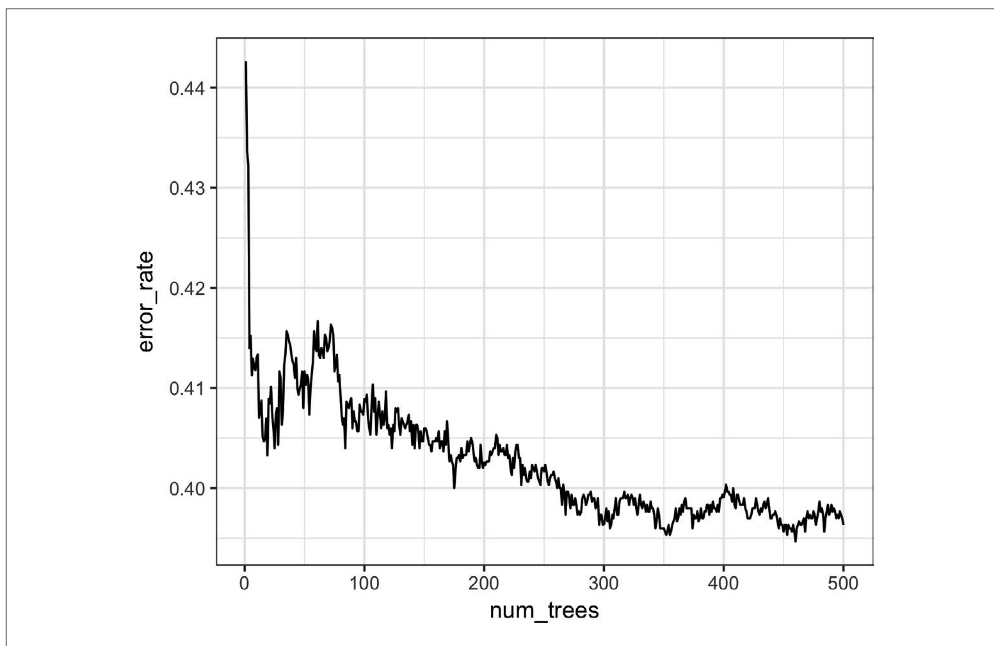


图 6-6：随着添加更多的树模型，随机森林的准确性得到了提升

图 6-7 很好地揭示了随机森林的本质。

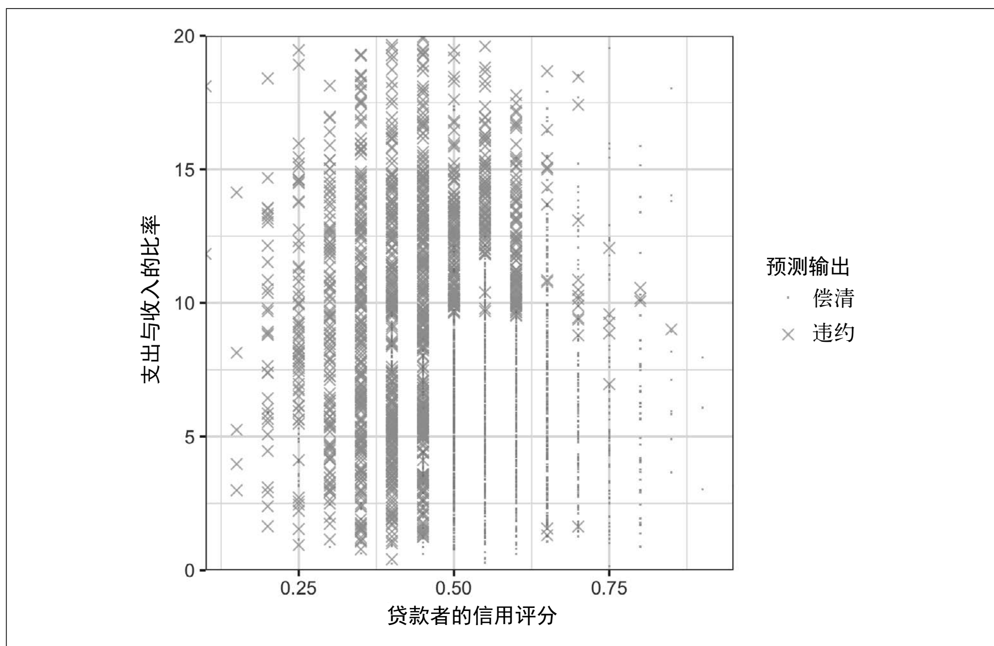


图 6-7：应用于贷款拖欠数据的随机森林的预测结果

随机森林方法是一种“黑箱”方法。与单个简单树模型相比，随机森林可以做出更准确的预测，但是丧失了单个树模型的直观决策规则。随机森林的预测也存在某种程度的噪声。我们可以注意到，一些借款者的分值非常高，这表明他们虽然具有较高的信用，但最终预测依然是贷款拖欠。这是由数据中的一些异常记录所导致的，也展示了随机森林过拟合的危险（参见 6.1.5 节中的知识点“偏差与方差的权衡”）。

6.3.3 变量的重要性

在为具有多个特征和记录的数据构建预测模型时，随机森林算法的强大得以尽显。该算法可以自动确定重要的预测变量，并可以发现交互项所对应的预测变量之间的复杂关系（参见 4.5.4 节）。例如，下面的代码在拟合模型时，使用了具有全部列的贷款拖欠数据。

```
> rf_all <- randomForest(outcome ~ ., data=loan_data, importance=TRUE)
> rf_all

Call:
 randomForest(formula = outcome ~ ., data = loan_data, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of error rate: 34.38%
Confusion matrix:
         paid off default class.error
paid off    15078    8058   0.3482884
default      7849   15287   0.3392548
```

参数 `importance=TRUE` 设置 `randomForest` 函数存储关于各个变量重要性的额外信息。函数 `varImpPlot` 绘制了变量的相对性能。

```
varImpPlot(rf_all, type=1)
varImpPlot(rf_all, type=2)
```

结果如图 6-8 所示。

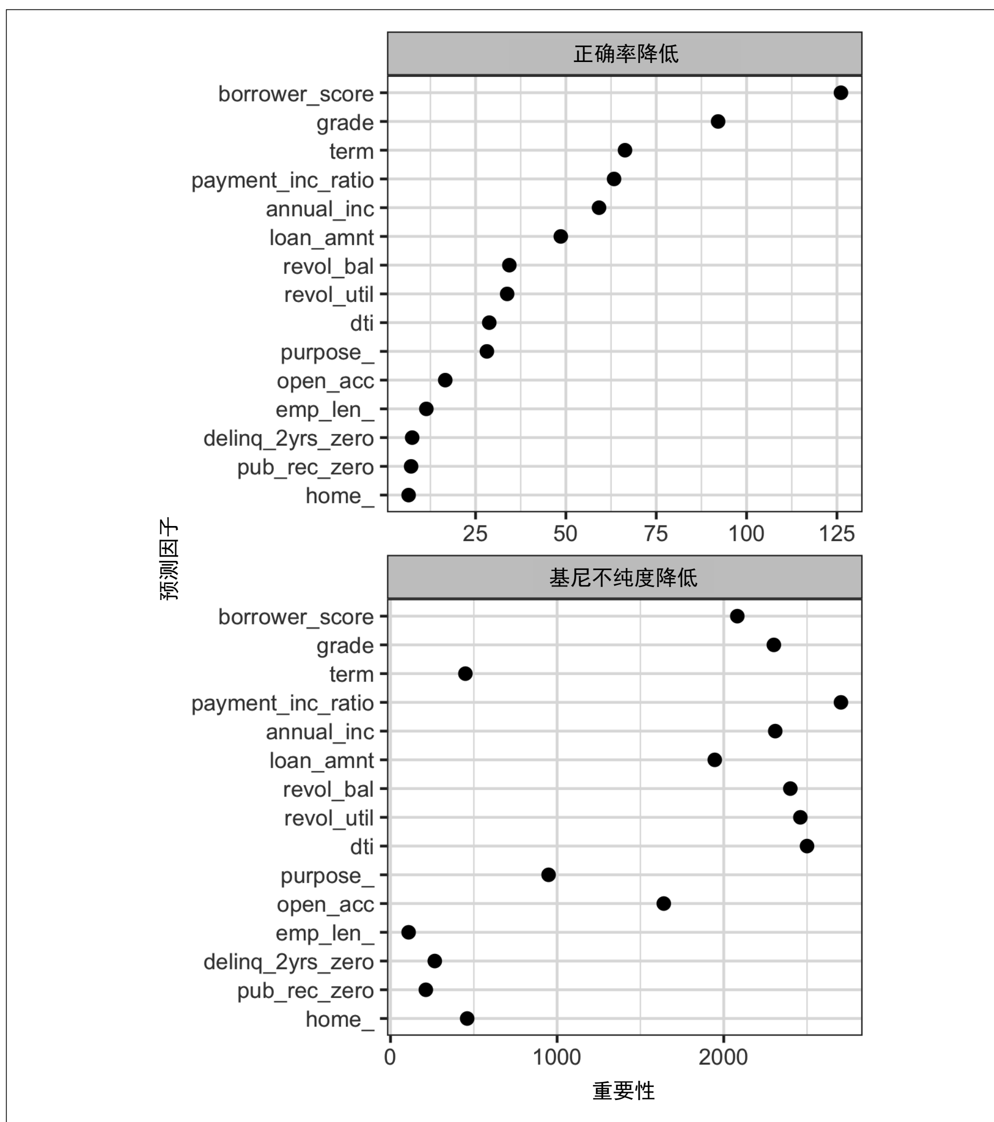


图 6-8：使用完整贷款数据拟合的模型，以及模型中变量的重要性

测定变量重要性有两种方法。

- 参数 `type=1`：如果一个变量的值是随机置换的，那么可以通过降低模型的正确率测定变量重要性。对变量值做随机置换，可以移除变量全部的预测能力。这时，可以从包外数据计算得到变量的重要性。（由此可见，这种测量方法实际上是一种交叉验证估计。）
- 参数 `type=2`：如果所有节点是使用某个变量拆分的，那么可以通过降低基尼不纯度分值的均值测定变量的重要性（参见 6.2.3 节）。这种方法测量了变量对于提升节点纯度的贡献度。它基于训练集，因此可靠性要低于由包外数据计算的度量。

图 6-8 的上图和下图分别显示了根据正确率的降低和基尼不纯度的降低这两种方法测得的变量重要性。两个图中的变量都是按正确率从高到低的顺序排列的。这两种测量方式得到的变量重要性分值完全不同。

既然正确率的降低是更可靠的度量，我们为什么还要使用基尼不纯度的降低这种测量呢？默认情况下，randomForest 函数只计算基尼不纯度。该算法一定会给出基尼不纯度，但是计算变量的模型正确率则需要额外进行一些计算，包括随机置换并预测数据。在计算的复杂度很重要的情况下，例如在需要拟合数千个模型的生产环境中，我们不应额外增加计算量。此外，基尼不纯度降低揭示了随机森林在生成拆分规则中使用了哪些变量（回想一下，该信息很容易在单个简单树模型中看到，但是在随机森林中就会丢失）。通过查看基尼不纯度降低与模型正确率这两种测定变量重要性方法之间的差异，有可能会找出一些改进模型的方法。

6.3.4 超参数

和许多统计机器学习算法一样，随机森林也可以看成一种“黑箱”算法，但是在箱子上会提供一些调节算法工作方式的旋钮。我们称这些旋钮为**超参数**。超参数是一些需要在拟合模型前设置的参数，它们并不会在算法训练过程中得到优化。虽然在传统的统计模型中，我们需要做出一些选择（例如在回归模型中需要选定预测变量），但是随机森林的超参数更为关键，尤其对于避免产生过拟合的问题。对于随机森林，我们需要介绍两个重要的超参数。

`nodesize`
: 末端节点（即树的叶子）的最小规模。对于分类，默认值为 1；对于回归，默认值为 5。

`maxnodes`
: 每个决策树中的最大节点数。默认情况下，没有限制。需拟合的最大树会受到 nodesize 的限制。

我们可能想忽略这些参数，只使用默认值。但是，如果对噪声数据应用随机森林，那么使用默认值可能会导致过拟合。当增大 nodesize 或设置 maxnodes 后，算法将拟合一个较小的树模型，而且不太可能给出假的预测规则。我们可以使用交叉验证（参见 4.2.3 节），检验设置不同超参数值的效果。

本节要点

- 通过组合多个模型的结果，集成模型提高了模型正确率。
- Bagging 是一类特殊的集成模型，它使用数据的自助法抽样拟合多个模型，并对模型取平均值。
- 随机森林是一种应用于决策树的特殊 Bagging 方法。除了对数据重抽样之外，随机森林算法还在拆分树时对预测变量做抽样。
- 对变量重要性的一种度量是随机森林的一种有用输出。变量重要性根据变量对模型正确率的贡献度，对变量排序。
- 随机森林具有一组超参数。可以使用交叉验证调整超参数，以避免产生过拟合。

6.4 Boosting

集成模型已经成为预测建模的一种标准工具。Boosting 是创建集成模型的一种通用方法。它与 Bagging（参见 6.3 节）几乎同时被提出。和 Bagging 一样，Boosting 也最常用于决策树。尽管两者之间具有很多相似之处，但是 Boosting 采用了完全不同的方法，实现也更为庞杂。因此，Bagging 可以在略微调整的情况下使用，但在应用 Boosting 时则需要更为慎重。如果我们用汽车来比喻这两种方法，那么 Bagging 可以看成本田雅阁，可靠并且稳定，而 Boosting 可以看成保时捷，尽管强大，但需要谨慎驾驶。

在线性回归模型中，经常需要检查残差的情况，以查看是否可以改进拟合（参见 4.6.4 节）。Boosting 更进一步，它对一系列模型做拟合，并使用当前的拟合模型去最小化之前模型的误差。Boosting 算法有几种常见的变体，分别是 Adaboost、**梯度提升**和**随机梯度提升**。其中，随机梯度提升最通用也是使用最广泛的。实际上，如果选择了正确的参数，Boosting 算法可以与随机森林相媲美。

主要术语

集成

使用一组模型做出预测。

同义词：模型平均

Boosting

在拟合一组模型时所使用的一种通用方法。Boosting 在每轮连续的拟合中，会对具有更大残差的记录赋予更大的权重。

Adaboost

Boosting 算法的一种早期实现，它根据残差的情况对数据重新加权。

梯度提升

一种更通用的 Boosting 算法。它将问题转化为代价函数最小化的问题。

随机梯度提升（SGD）

最常用的 Boosting 算法。它在每轮拟合中加入了对记录和数据列的重抽样。

正则化

通过在代价函数中对模型参数的数量添加惩罚项，避免产生过拟合。

超参数

在拟合算法之前就需要设定的参数。

6.4.1 Boosting算法

各种 Boosting 算法背后的基本理念大致相同。其中，Adaboost 算法是最易于理解的，该算法的步骤如下。

(1) 初始化要拟合模型的最大数量 M，并设置迭代计数器 $m = 1$。初始化观测值权重 $w_i = \frac{1}{N}$，其中 $i = 1, 2, \cdots, N$。初始化集成模型 $\hat{F}_0 = 0$。

(2) 使用 $\hat{f}_m$ 训练模型。其中，$\hat{f}_m$ 使用使加权误差 e_m 最小化的观测权重 $w_1, w_2, \cdots, w_N$。加权误差 e_m 定义为误分类观测的权重总和。

(3) 将模型添加到集成中：$\hat{F}_m = \hat{F}_{m-1} + \alpha_m \hat{f}_m$，其中 $\alpha_m = \log(\frac{1-e_m}{e_m})$。

(4) 更新权重 $w_1, w_2, \cdots, w_N$，使误分类观测值的权重增加。权重增加的规模取决于 α_m。α_m 值越大，权重增加得越大。

(5) 增加模型计数器，$m = m + 1$。如果 $m \leqslant M$，返回步骤 1。

Boosting 算法的估计值由下式给出：

$$\hat{F} = \alpha_1 \hat{f}_1 + \alpha_2 \hat{f}_2 + \cdots + \alpha_M \hat{f}_M$$

Boosting 算法通过增加误分类观测值的权重，强制模型在训练中偏重于表现不佳的数据。因子 α_m 确保了误差较低的模型具有更大的权重。

梯度提升算法与 Adaboost 类似，只是它将问题转化为代价函数的优化问题。梯度提升并不是调整权重，而是根据**伪残差**去拟合模型，这使得在训练中偏重于较大的残差。梯度随机提升遵循随机森林的理念，即在每一个阶段对观测值和预测变量抽样，由此在算法中加入了随机性。

6.4.2　XGBoost软件

XGBoost 是最广为使用的 Boosting 公共域软件，它是随机梯度提升算法的一种实现，最初是由华盛顿大学的陈天奇和 Carlos Guestrin 开发的。XGBoost 是一种具有多种选项的计算很高效的实现，并在大多数主流数据科学语言中以软件包形式提供。在 R 语言中，提供 XGBoost 的软件包是 `xgboost`。

软件包中的函数 `xgboost` 提供了多个可调整并且应该调整的参数（参见 6.4.4 节）。其中，两个非常重要的参数是 `subsample` 和 `eta`。`subsample` 参数控制每次迭代时应该被抽样的部分观测值；`eta` 设置了 Boosting 算法中 α_m 的收缩因子（参见 6.4.1 节）。设置 `subsample` 参数可以使 Boosting 算法类似随机森林，只是在完成抽样后不放回样本。收缩因子 `eta` 通过降低权重的变化，防止产生过拟合（权重变化较小，意味着算法更不易于过拟合训练数据）。在下面的代码中，我们对只有两个预测变量的贷款数据应用 `xgboost`。

```
library(xgboost)
predictors <- data.matrix(loan3000[, c('borrower_score',
                                       'payment_inc_ratio')])
label <- as.numeric(loan3000[,'outcome'])-1
xgb <- xgboost(data=predictors, label=label,
               objective = "binary:logistic",
               params=list(subsample=.63, eta=0.1), nrounds=100)
```

注意，`xgboost` 不支持公式语法，因此需要将预测变量转换为 R 语言的 `data.matrix` 对象，而响应变量需要转换为 0/1 二元变量。参数 `objective` 指定了 `xgboost` 函数所处理的问题类型。`xgboost` 根据该参数选取优化指标。

算法的预测值可以使用 predict 函数得到。由于在本例中只使用了两个变量，因此只绘制预测变量和预测值。

```
pred <- predict(xgb, newdata=predictors)
xgb_df <- cbind(loan3000, pred_default=pred>.5, prob_default=pred)
ggplot(data=xgb_df, aes(x=borrower_score, y=payment_inc_ratio,
                        color=pred_default, shape=pred_default)) +
       geom_point(alpha=.6, size=2)
```

结果如图 6-9 所示。从定性的角度来看，算法的输出类似于图 6-7 中随机森林做出的预测。在预测中存在一些噪声，因为有些借贷信用评分很高的借款者仍被预测为拖欠。

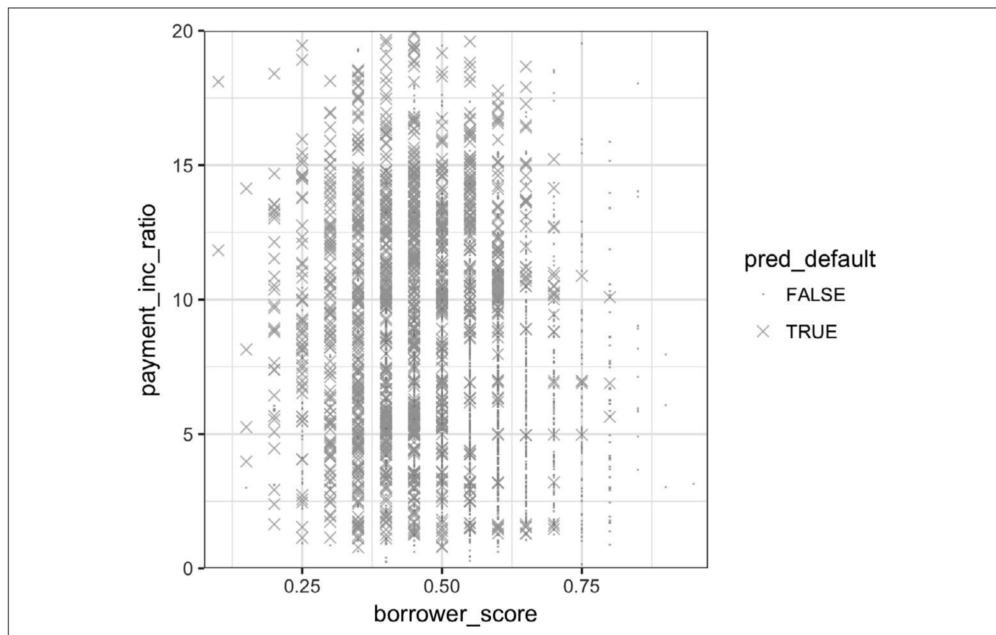


图 6-9：将 XGBoost 应用于贷款拖欠数据得到的预测结果

6.4.3 正则化：避免过拟合

如果盲目地应用 xgboost 函数，可能会**过拟合**训练数据，进而导致模型不稳定。过拟合问题是两方面的。

- 应用于不在训练集中的新数据时，模型的正确率会降低。
- 模型给出的预测是高度可变的，导致了结果不稳定。

任何建模技术都可能产生过拟合。例如，如果回归方程中包含了过多的变量，那么模型可能会给出虚假预测。然而对于大多数统计技术而言，可以通过审慎地选择预测变量来避免产生过拟合。即便是随机森林，通常也能给出一个合理的模型，而无须调整参数。但这并不适用于 xgboost。下面的代码在训练数据上使用 xgboost 拟合贷款数据，模型中考虑了全部变量。

```
> predictors <- data.matrix(loan_data[,-which(names(loan_data) %in%
                                       'outcome')])
> label <- as.numeric(loan_data$outcome)-1
> test_idx <- sample(nrow(loan_data), 10000)
> xgb_default <- xgboost(data=predictors[-test_idx,],
                         label=label[-test_idx],
                         objective = "binary:logistic", nrounds=250)
> pred_default <- predict(xgb_default, predictors[test_idx,])
> error_default <- abs(label[test_idx] - pred_default) > 0.5
> xgb_default$evaluation_log[250,]
   iter train_error
1:  250    0.145622
> mean(error_default)
[1] 0.3715
```

测试集由从完整数据集中随机抽取的 10 000 条记录构成，而训练集则由剩余的记录构成。Boosting 算法在训练集上的错误率仅约为 14.6%，而在测试集上的错误率则高得多，约为 37.2%。这是由过拟合导致的。Boosting 算法虽然很好地解释了训练集的变异性，但是它生成的预测规则并不适用于新数据。

为了避免产生过拟合，Boosting 算法需要设置多个参数，其中包括上一节中介绍的 `eta` 和 `subsample`。另一种方法是使用**正则化**。正则化是一种通过修改代价函数去**惩罚**模型复杂度的技术。在决策树的拟合中，使用最小化基尼不纯度分值（参见 6.2.3 节）等代价准则。在 `xgboost` 中，可以通过添加一个衡量模型复杂度的项，实现对代价函数的修改。

`xgboost` 提供了两个用于模型正则化的参数：`alpha` 和 `lambda`，它们分别表示曼哈顿距离和欧氏距离的平方（参见 6.1.2 节）。增大这两个参数将会惩罚更复杂的模型，并减小拟合树模型的规模。例如，下面的代码探索了将 `lambda` 设置为 1000 的情况。

```
> xgb_penalty <- xgboost(data=predictors[-test_idx,],
                         label=label[-test_idx],
                         params=list(eta=.1, subsample=.63, lambda=1000),
                         objective = "binary:logistic", nrounds=250)
> pred_penalty <- predict(xgb_penalty, predictors[test_idx,])
> error_penalty <- abs(label[test_idx] - pred_penalty) > 0.5
> xgb_penalty$evaluation_log[250,]
   iter train_error
1:  250    0.332405
> mean(error_penalty)
[1] 0.3483
```

现在，我们得到的训练误差仅略低于测试集上的误差。

函数 `predict` 提供了一个便利的参数 `ntreelimit`，该参数强制在预测中仅使用前 i 个树模型。这让我们在添加更多的模型时，可以直接比较样本内和样本外的错误率。

```
> error_default <- rep(0, 250)
> error_penalty <- rep(0, 250)
> for(i in 1:250){
  pred_def <- predict(xgb_default, predictors[test_idx,], ntreelimit=i)
  error_default[i] <- mean(abs(label[test_idx] - pred_def) >= 0.5)
  pred_pen <- predict(xgb_penalty, predictors[test_idx,], ntreelimit = i)
  error_penalty[i] <- mean(abs(label[test_idx] - pred_pen) >= 0.5)
}
```

在模型输出的 xgb_default$evaluation_log 项中，返回了训练集上的误差。结合样本外误差，我们就可以绘制误差与迭代次数的关系。

```
> errors <- rbind(xgb_default$evaluation_log,
                  xgb_penalty$evaluation_log,
                  data.frame(iter=1:250, train_error=error_default),
                data.frame(iter=1:250, train_error=error_penalty))
> errors$type <- rep(c('default train', 'penalty train',
                       'default test', 'penalty test'), rep(250, 4))
> ggplot(errors, aes(x=iter, y=train_error, group=type)) +
  geom_line(aes(linetype=type, color=type))
```

结果如图 6-10 所示。该图展示了默认模型在训练集上的预测准确性稳定地提高了，但在测试集上的准确性却降低了。惩罚模型并不会表现出这种行为。

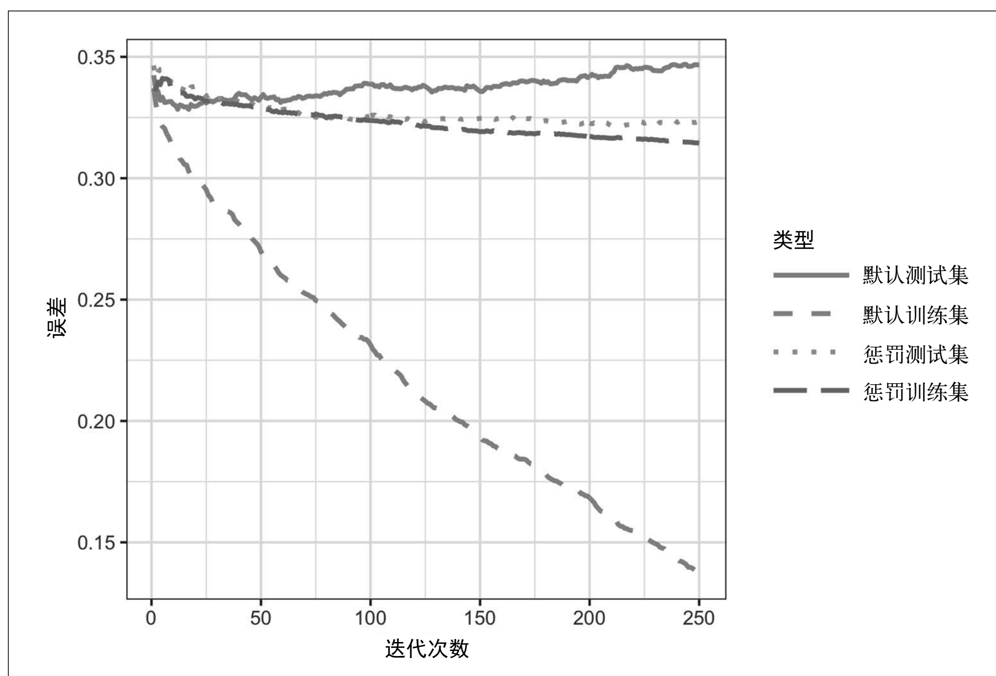


图 6-10：默认 XGBoost 与惩罚 XGBoost 的错误率对比

岭回归和 Lasso 回归

对模型的复杂性添加惩罚项，有助于避免产生过拟合。该理念可追溯至 20 世纪 70 年代。最小二乘回归会最小化残差平方和（RSS），参见 4.1.3 节。而**岭回归**则最小化残差平方和，并对系数的数量和大小添加惩罚项：

$$\sum_{i=1}^{n}(Y_i-\hat{b}_0-\hat{b}_1X_i-\cdots-\hat{b}X_p)^2+\lambda(\hat{b}_1^2+\cdots+\hat{b}_p^2)$$

> λ 的值决定了系数被惩罚的程度。由较大的值所生成的模型不太可能过拟合数据。**Lasso 回归**与此类似，只是其惩罚项使用的是曼哈顿距离，而非欧氏距离：
>
> $$\sum_{i=1}^{n}(Y_i-\hat{b}_0-\hat{b}_1X_i-\cdots-\hat{b}X_p)^2+\alpha(|\hat{b}_1|+\cdots+|\hat{b}_p|)$$
>
> 前面介绍的 xgboost 参数 lambda 和 alpha 具有类似的作用。

6.4.4　超参数和交叉验证

xgboost 函数具有一组令人望而却步的超参数，相关介绍可参见本节的知识点“XGBoost 超参数”。正如 6.4.3 节中介绍的，特定的参数选择可以显著地改进模型拟合。鉴于超参数有大量的组合，我们应该如何从中做出选择呢？这个问题的标准解决方案是使用**交叉验证**，参见 4.2.3 节。交叉验证将数据随机拆分为 K 个不同的组，也被称为 K 折。对于每折，交叉验证使用非折内数据训练模型，然后使用折内数据评估模型。这为模型在样本外数据上的正确率提供了一种度量。最佳的一组超参数可以由总体误差最低的模型给出，总体误差是通过对每折的误差取平均计算得到的。

为了介绍这一技术，我们将其应用于 xgboost 函数的参数选择。在本例中，我们使用了两个参数：收缩参数 eta（参见 6.4.2 节）和表示树模型最大深度的参数 max_depth。max_depth 定义了叶节点到树根的最大深度，默认值为 6。这提供了另一种控制过拟合的方式，即层数更多的树模型往往更为复杂，并可能会过拟合数据。首先，我们设置折数和参数列表。

```
> N <- nrow(loan_data)
> fold_number <- sample(1:5, N, replace = TRUE)
> params <- data.frame(eta = rep(c(.1, .5, .9), 3),
                       max_depth = rep(c(3, 6, 12), rep(3,3)))
```

然后我们应用前面介绍的算法，使用 5 折计算每个模型和每折的误差。

```
> error <- matrix(0, nrow=9, ncol=5)
> for(i in 1:nrow(params)){
>   for(k in 1:5){
>     fold_idx <- (1:N)[fold_number == k]
>     xgb <- xgboost(data=predictors[-fold_idx,], label=label[-fold_idx],
                     params = list(eta = params[i, 'eta'],
                                   max_depth = params[i, 'max_depth']),
                   objective = "binary:logistic", nrounds=100, verbose=0)
>     pred <- predict(xgb, predictors[fold_idx,])
>     error[i, k] <- mean(abs(label[fold_idx] - pred) >= 0.5)
>   }
> }
```

由于一共需要拟合 45 个模型，因此代码可能会运行一段时间。误差存储在一个矩阵对象中，矩阵的行是模型，列为折数。我们可以使用函数 rowMeans 比较各个参数集的错误率。

```
> avg_error <- 100 * rowMeans(error)
> cbind(params, avg_error)
  eta max_depth avg_error
```

```
  1 0.1         3     35.41
  2 0.5         3     35.84
  3 0.9         3     36.48
  4 0.1         6     35.37
  5 0.5         6     37.33
  6 0.9         6     39.41
  7 0.1        12     36.70
  8 0.5        12     38.85
  9 0.9        12     40.19
```

交叉验证指出，使用了较小的 eta 值并且层数较少的树模型，会得出更准确的结果。由于这样的模型也更稳定，所以应使用的最佳参数是 eta = 0.1，max_depth = 3（也可能是 max_depth = 6）。

XGBoost 超参数

xgboost 的超参数主要用于在过拟合与正确率和计算复杂度之间取得平衡。关于 XGBoost 超参数的完整介绍，参见 xgboost 的文档。

eta
: 值位于 0 和 1 之间的收缩因子，即 Boosting 算法的 α。默认值为 0.3。但是对于噪声数据，推荐使用更小的值，例如 0.1。

nrounds
: 设置 Boosting 算法的循环次数。如果将 eta 设置为一个较小的值，这会增加循环的次数，因为算法学习的速度更慢。只要在计算中设置了一些防止过拟合的参数，那么运行更多轮循环也没问题。

max_depth
: 设置树模型的最大深度，默认值为 6。与随机森林拟合非常深的树模型不同，Boosting 算法通常会拟合一个层数不多的树模型。这样做的优点是，可以避免由噪声数据导致模型中出现虚假的复杂交互。

subsample 和 colsample_bytree
: subsample 指定了做无放回抽样的部分记录，colsample_bytree 指定了在拟合树模型中要抽样的部分预测变量。这些参数类似于随机森林中使用的相应参数，有助于避免产生过拟合。

lambda 和 alpha
: 帮助控制过拟合的正则化参数（参见 6.4.3 节）。

本节要点

- Boosting 是一类基于对一组模型做拟合的集成模型。在连续的每轮拟合中，Boosting 算法会为具有更大残差的记录赋予更大的权重。
- 随机梯度提升是最通用的 Boosting 算法，具有最佳性能。随机梯度提升最常见的形式是使用树模型。

- XGBoost 是一种广为使用的随机梯度提升软件包，它计算高效。所有数据科学常用的语言中都提供了 XGBoost。
- Boosting 容易过拟合数据。为了避免产生过拟合，需要调整超参数。
- 正则化通过在模型的参数数量（例如，树的规模）上添加惩罚项，避免产生过拟合。
- 鉴于 Boosting 算法需要设置大量的超参数，交叉验证尤为重要。

6.5 小结

本章介绍了两种分类和预测方法，它们并非从拟合整个数据集的结构化模型（如线性回归）着手，而是从数据本身进行灵活、局部的学习。*K* 最近邻是一个简单的过程，它只查看相似的记录，并将预测记录指定为相似记录中的多数类（或均值）。树模型探索预测变量的各种截止值（拆分值），用于将数据迭代地拆分为分类逐步趋同的分区和子分区。最有效的拆分值会构成一条用于分类或预测的路径，或称为“规则”。树模型是一类非常强大的预测工具，通常会给出优于其他方法的结果，因此得到了广泛的使用。树模型催生了多种集成方法，例如随机森林、Boosting、Bagging 等，它们提高了树模型的预测能力。

第 7 章

无监督学习

无监督学习指的是无须使用已标记数据（即输出已知的数据）训练模型，便可以抽取数据内涵的统计学方法。在第 4 章和第 5 章中，我们的目标是使用一组预测变量构建一个用于预测响应变量的模型（规则集合）。无监督学习也对数据建模，但并不区分响应变量和预测变量。

无监督学习可以有多种目标。在一些情况下，无监督学习方法可用于对缺少有标记响应的数据创建预测规则。**聚类**方法可用于识别数据中有意义的分组。例如，使用一个网站的 Web 点击数和用户统计数据，可以对不同类型的用户进行分组，进而根据各组用户的特性实现网站的个性化。

在另一些情况下，无监督学习的目标可能是将数据**降维**至可管理的一组变量，进而将降维的变量集作为回归或分类等预测模型的输入。例如，在工业流程中可能要监控数千个传感器。我们或许不需要考虑每个传感器的数据流，仅通过将数据规约为一个更小的特征集，就能建立一个更强大并可解释的模型，预测一个流程是否会失败。

最后，我们可以将无监督学习看成为了适应海量变量和记录，而对探索性数据分析（参见第 1 章）做的一个扩展。其目的是深入探索一组数据及其中各个变量之间的关系。使用无监督技术可以筛选并分析这些变量，进而发现各变量间的关系。

无监督学习和预测

在回归和分类等预测问题中，无监督学习可以发挥重要作用。在某些情况下，我们希望能在缺少标记数据的情况下预测一个类别。例如，我们可能想使用一组卫星传感数据，预测某个地区的植被类型。鉴于我们并没有响应变量去训练模型，这时可使用聚类方法识别其中的常见模式，并对地理区域进行分类。

聚类是一种解决“冷启动问题”的尤为重要的工具。在冷启动问题中，例如推出新的营销活动、识别潜在的新型欺诈或垃圾邮件，可能一开始我们并没有任何响应可用于训练模型。随着时间的推移和数据的积累，我们才有可能更多地了解系统，并构建传统的预测模型。但是，聚类通过确定如何拆分总体，让我们可以更快地开始学习过程。

无监督学习也是回归和分类技术的重要组成部分。对于大数据而言，如果总体中的某个子集并不具有代表性，那么训练好的模型可能会在该子集上表现不佳。聚类可以识别并标记子集，进而对各个子集拟合不同的模型。或者，也可以用一些特定的特征去表示子集，并强制整个模型明确地以所识别的子集为预测因子。

7.1 主成分分析

变量常常会一起发生变化（共变）。实际上，不同变量的部分变化可能会重叠。**主成分分析**（PCA）就是一种能够发现数值型变量共变方式的技术[1]。

主要术语

主成分

预测变量的一种线性组合。

载荷

将预测因子转换为成分的过程中所使用的权重值。

同义词：权重

陡坡图

一种展示各成分方差的绘图，图中显示了各成分的相对重要性。

主成分分析的基本理念是，将多个数值型预测变量组合成一组规模较小的变量，它们是原始变量的加权线性组合。所形成的规模较小的一组变量被称为**主成分**。主成分可以“解释”完整变量集的大部分变异性，同时降低数据维度。在构建主成分中所使用的权重，体现了原始变量对新的主成分的相对贡献。

注 1：本章中各节的内容，版权属于本书作者彼得·布鲁斯和安德鲁·布鲁斯，© 2017 Datastats, LLC。使用需经许可。

主成分分析最早是由卡尔·皮尔逊（Karl Pearson）在 1901 年发表的一篇论文中提出的。这篇论文可能是首篇公开发表的无监督学习论文。皮尔逊在论文中指出，在许多问题中，预测变量存在变异性，因此他提出了对该变异性建模的主成分分析技术。主成分分析可以看成一种无监督版的线性判别分析（参见 5.2 节）。

7.1.1　一个简单的例子

对于两个变量 X_1 和 X_2，具有两个主成分 Z_i（$i = 1, 2$）：

$$Z_i = w_{i,1}X_1 + w_{i,2}X_2$$

其中，权重 $w_{i,1}$ 和 $w_{i,2}$ 被称为成分**载荷**，用于将原始变量转换为主成分。第一主成分 Z_1 最好地解释了总变异性的线性组合。第二主成分 Z_2 解释了剩余的变异性，它也是最差拟合的线性组合。



还有一种常用的主成分计算方法。该方法使用了预测变量与均值之间的偏离情况，而非预测变量本身。

在 R 语言中，可以使用函数 `princomp` 计算主成分。下面的代码对雪佛龙公司（CVX）和埃克森美孚公司（XOM）的股票收益做了主成分分析。

```
oil_px <- sp500_px[, c('CVX', 'XOM')]
pca <- princomp(oil_px)
pca$loadings

Loadings:
    Comp.1 Comp.2
CVX -0.747  0.665
XOM -0.665 -0.747
```

对于 CVX 和 XOM 的股票收益数据，第一主成分的权重分别是 −0.747 和 −0.665，第二主成分的权重分别是 0.665 和 −0.747。这个结果应该如何解释？第一主成分基本上是 CVX 和 XOM 的平均值，反映了这两家能源公司之间的相关性。第二主成分则测定了两支股票的价格何时出现偏离。

绘制数据与主成分，也有一定的指导意义。

```
loadings <- pca$loadings
ggplot(data=oil_px, aes(x=CVX, y=XOM)) +
  geom_point(alpha=.3) +
  stat_ellipse(type='norm', level=.99) +
  geom_abline(intercept = 0, slope = loadings[2,1]/loadings[1,1]) +
  geom_abline(intercept = 0, slope = loadings[2,2]/loadings[1,2])
```

结果如图 7-1 所示。

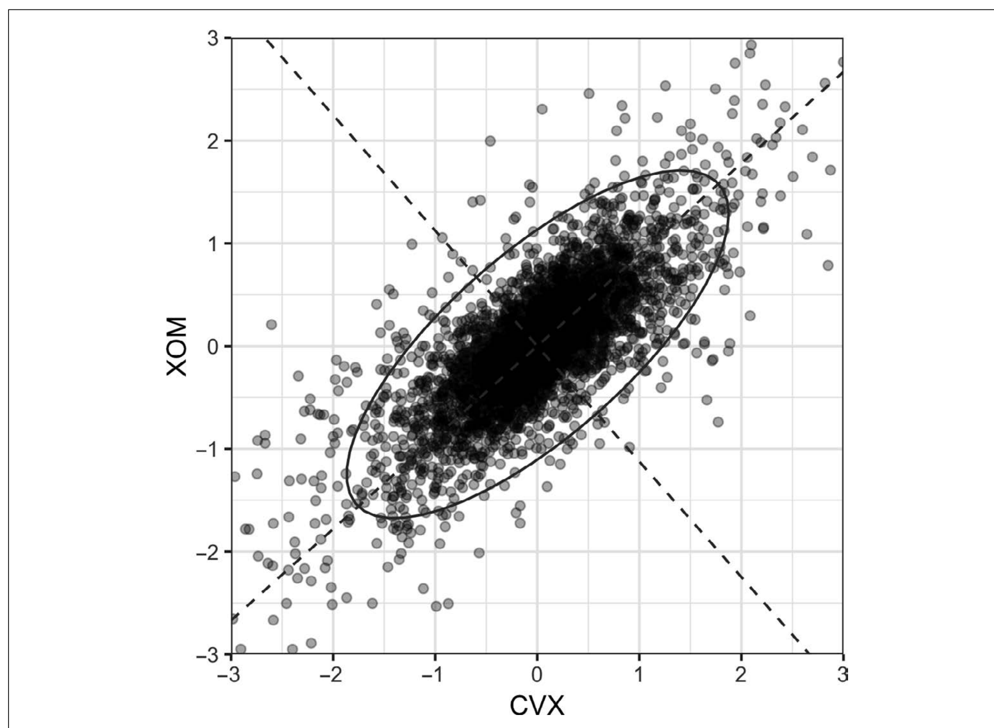


图 7-1：CVX 和 XOM 股票收益的主成分

虚线表示了两个主成分。第一主成分沿椭圆的长轴，第二主成分沿椭圆的短轴。我们可以看到，两支股票收益的主要变化是由第一主成分解释的。这是合理的，因为能源股的价格倾向于整体发生变化。



从上例中我们看到，第一主成分的权重均为负值。将所有权重的正负值反转，并不会改变主成分。例如，如果对第一主成分使用 0.747 和 0.665 的权重，等价于使用 −0.747 和 −0.665 的权重。这从图中也可以看出，经过原点和 (1, 1) 的无限长直线，等同于经过原点和 (−1, −1) 的无限长直线。

7.1.2　计算主成分

将主成分分析从两个变量扩展到多个变量是十分简单的。对于第一主成分，只需在线性组合中额外添加预测变量，并指定可优化所有预测变量共变集（在统计学中这被称为**协方差**，参见 5.2.1 节）的权重。主成分的计算使用了经典的统计学方法，依赖于数据的相关矩阵或协方差矩阵，并且计算执行得很快，不需要做迭代。如上节所述，主成分分析只适用于数值型变量，并不适用于分类变量。完整的计算过程如下。

(1) 在创建第一主成分时，主成分分析给出的预测变量的线性组合，使得可解释的总方差的比例最大化。

(2) 进而，主成分分析将该线性组合作为第一个“新的”预测因子 Z_1。
(3) 主成分分析使用具有不同权重的同一变量，重复上述过程，创建第二个“新的”预测因子 Z_2。主成分分析对两个预测因子加权，使得 Z_1 和 Z_2 不相关。
(4) 重复上述过程，直到所得到的新变量（或成分）Z_i 的数量与原始变量 X_i 相同。
(5) 选择保留为解释大部分方差所需的成分。
(6) 目前得到的结果是对应于每个成分的一组权重。最后一步是通过对原始数据应用权重，将原始数据转换为新的主成分分值。这些新的分值可以作为规模缩减的一组预测变量。

7.1.3 解释主成分

主成分的性质往往能揭示数据的结构信息。一些标准的可视化展示有助于我们深入了解主成分的相关信息。一种可视化方法就是**陡坡图**（Screeplot）。陡坡图展示了各个主成分的相对重要性。其命名源于其类似于山体的陡坡。使用下面的代码可以绘制在标准普尔 500 指数中排名靠前的几家公司的陡坡图。

```
syms <- c( 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX', 'XOM',
   'SLB', 'COP', 'JPM', 'WFC', 'USB', 'AXP', 'WMT', 'TGT', 'HD', 'COST')
top_sp <- sp500_px[row.names(sp500_px)>='2005-01-01', syms]
sp_pca <- princomp(top_sp)
screeplot(sp_pca)
```

结果如图 7-2 所示。从图中可见，第一主成分的方差相当大（通常情况下都会如此），但是其他几个靠前的主成分也是显著的。

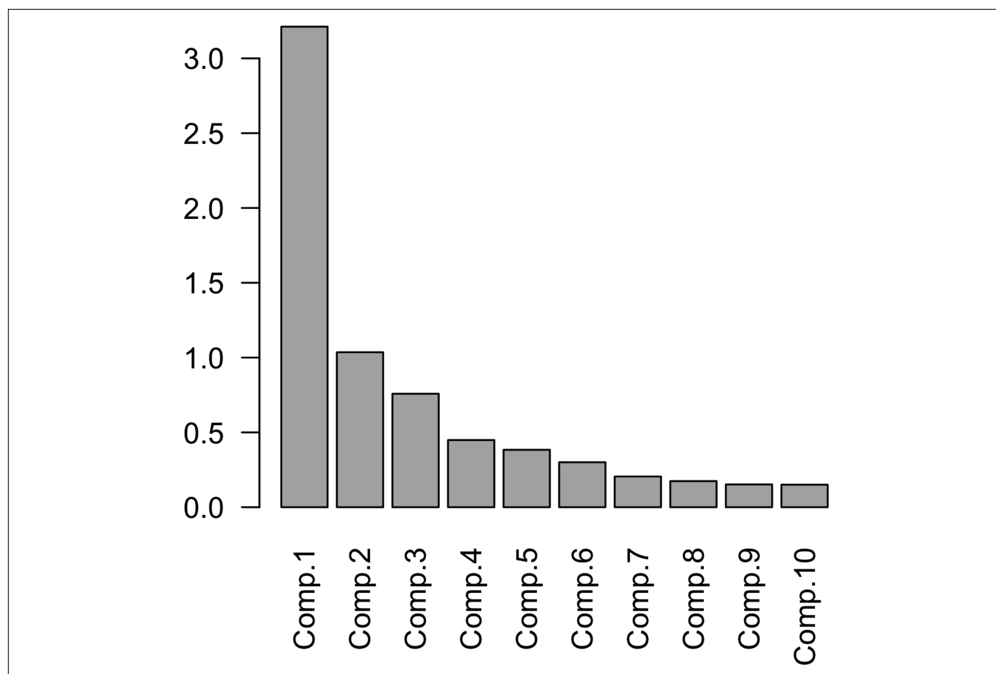


图 7-2：标准普尔 500 指数中排名靠前的几家公司股票的主成分分析陡坡图

如果绘制出前几个主成分的权重，那么绘图将更有启示作用。一种绘图方法是在 ggplot 中结合使用 tidyr 包中的 gather 函数。

```
library(tidyr)
loadings <- sp_pca$loadings[,1:5]
loadings$Symbol <- row.names(loadings)
loadings <- gather(loadings, "Component", "Weight", -Symbol)
ggplot(loadings, aes(x=Symbol, y=Weight)) +
  geom_bar(stat='identity') +
  facet_grid(Component ~ ., scales='free_y')
```

前 5 个主成分的载荷如图 7-3 所示。其中，第一主成分的载荷具有一致的正负。这对于所有列共享同一因子（在本例中是股市的整体趋势）的数据来说是一种典型情况。第二主成分捕获了能源类股票价格与其他类股票相比的变化情况。第三主成分主要对比了苹果公司（AAPL）和好市多（CostCo）股票的变化。第四主成分对比了斯伦贝谢公司（SLB）与其他能源类股票的走势。第五主成分由金融类公司主导。

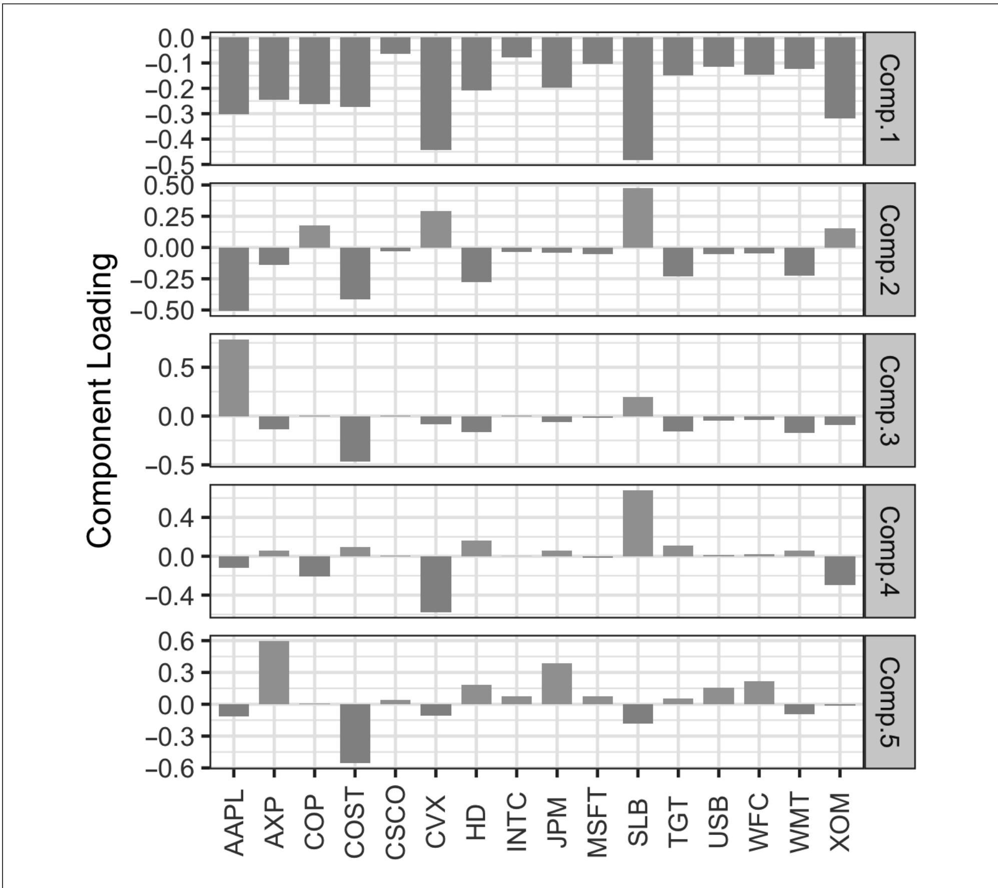


图 7-3：股价收益的前 5 个主成分的载荷



应选取多少个成分？

如果我们的目标是降低数据的维度，那么必须确定要选取的主成分数。最常用的方法是使用**即席**（ad hoc）规则，即选择解释了“大部分”方差的成分。我们可以通过陡坡图以可视化方式进行。例如，在图 7-2 中，很自然地要将分析限定于前五个成分。另一种方法是选择累积方差超过一定阈值（例如 80%）的前几个成分。此外，我们还可以通过检查载荷的情况，确定各个成分是否能给出直观的解释。对于选择重要成分的数量，一种正式的做法是使用交叉验证（更多信息，参见 4.2.3 节）。

本节要点

- 主成分是预测变量的线性组合，但仅限于数值型预测变量。
- 主成分计算的原则是使成分之间的相关性最小化，进而减少冗余。
- 通常，有限数量的成分就可以解释结果变量的大部分方差。
- 这样就可以使用一组有限的主成分代替（更多的）原始预测变量，从而降低维度。

7.1.4　拓展阅读

Rasmus Bro、Karin Kjeldahl、A. K. Smilde 和 Henk A. L. Kiers 于 2008 年发表在 *Analytical and Bioanalytical Chemistry* 期刊上的论文“Cross-Validation of Component Models: A Critical Look at Current Methods”详细地介绍了如何在主成分分析中使用交叉验证方法。

7.2　*K*-Means聚类

聚类是一种数据分组技术，使得组内记录彼此相似。聚类的一个目标是识别数据中重要且有意义的组。这些组可以直接使用，进行更深入的分析，也可以作为特征或结果传递给预测回归或分类模型。*K*-Means 是首个被提出来的聚类方法，目前依然广为使用，原因在于其算法相对简单，并可以扩展到大规模数据集上。

主要术语

类（cluster）
　　一组类似的记录。

类均值
　　表示类内记录变量均值的向量。

K
　　类的个数。

我们称一个类内每个记录到该类**均值**之间距离的平方和为**类内平方和**，或简称为“类内 SS”。*K*-Means 通过最小化类内平方和，将数据划分为 *K* 个类。*K*-Means 并不能保证各个类的规模相同，但是能找出相互分离情况最好的类。



归一化

应对连续变量做归一化（标准化），一般做法是减去均值再除以标准偏差。否则大尺度的变量将会主导聚类过程（参见 6.1.4 节）。

7.2.1　一个简单的例子

给定一个具有 *n* 条记录的数据集，其中只有 *x* 和 *y* 两个变量。假设我们想将数据划分为 $K = 4$ 个类，这意味着每条记录 (x_i, y_i) 将会指定给一个类 *k*。假定给类 *k* 指定了 n_k 条记录，类的中心 $(\bar{x}_k, \bar{y}_k)$ 就是类内各记录的均值，即：

$$\bar{x}_k = \frac{1}{n_k} \sum_{i \in 类k} x_i$$

$$\bar{y}_k = \frac{1}{n_k} \sum_{i \in 类k} y_i$$



类均值

通常情况下，聚类的记录具有多个变量。**类均值**（cluster mean）并非指单个数字，而是指表示变量均值的向量。

类内平方和由下式给出。

$$\mathrm{SS}_k = \sum_{i \in 类k} (x_i - \bar{x}_k)^2 + (y_i - \bar{y}_k)^2$$

K-Means 将给出一种记录的分配方法，使得所有四个类的类内平方和之和 $\mathrm{SS}_1 + \mathrm{SS}_2 + \mathrm{SS}_3 + \mathrm{SS}_4$ 最小化。

$$\sum_{k=1}^{4} \mathrm{SS}_i$$

使用 *K*-Means 聚类，我们可以深入地了解股票价格走势的聚类情况。注意，股票收益实际上是以归一化的方式给出的，因此不需要再对数据做归一化。在 R 语言中，可以使用 `kmeans` 函数执行 *K*-Means 聚类。例如，下面的代码使用了 XOM 和 CVX 股票收益这两个变量，划分出 4 个类。

```
df <- sp500_px[row.names(sp500_px)>='2011-01-01', c('XOM', 'CVX')]
km <- kmeans(df, centers=4)
```

在代码的输出结果中，每个记录的聚类分配情况在 `cluster` 项中。

```
> df$cluster <- factor(km$cluster)
> head(df)
                  XOM        CVX cluster
2011-01-03 0.73680496  0.2406809       2
2011-01-04 0.16866845 -0.5845157       1
2011-01-05 0.02663055  0.4469854       2
2011-01-06 0.24855834 -0.9197513       1
2011-01-07 0.33732892  0.1805111       2
2011-01-10 0.00000000 -0.4641675       1
```

从结果中可以看到，前 6 条记录被指定给了类 1 或类 2。此外，代码还返回了类均值。

```
> centers <- data.frame(cluster=factor(1:4), km$centers)
> centers
  cluster        XOM        CVX
1       1 -0.3284864 -0.5669135
2       2  0.2410159  0.3342130
3       3 -1.1439800 -1.7502975
4       4  0.9568628  1.3708892
```

类 1 和类 3 代表股市“走低”，而类 2 和类 4 表示股市“上涨”。在本例中，只有两个变量，所以可以直观地查看各个类及其含义。

```
ggplot(data=df, aes(x=XOM, y=CVX, color=cluster, shape=cluster)) +
  geom_point(alpha=.3) +
  geom_point(data=centers,  aes(x=XOM, y=CVX), size=3, stroke=2)
```

结果如图 7-4 所示，图中显示了类的分配情况和类均值。

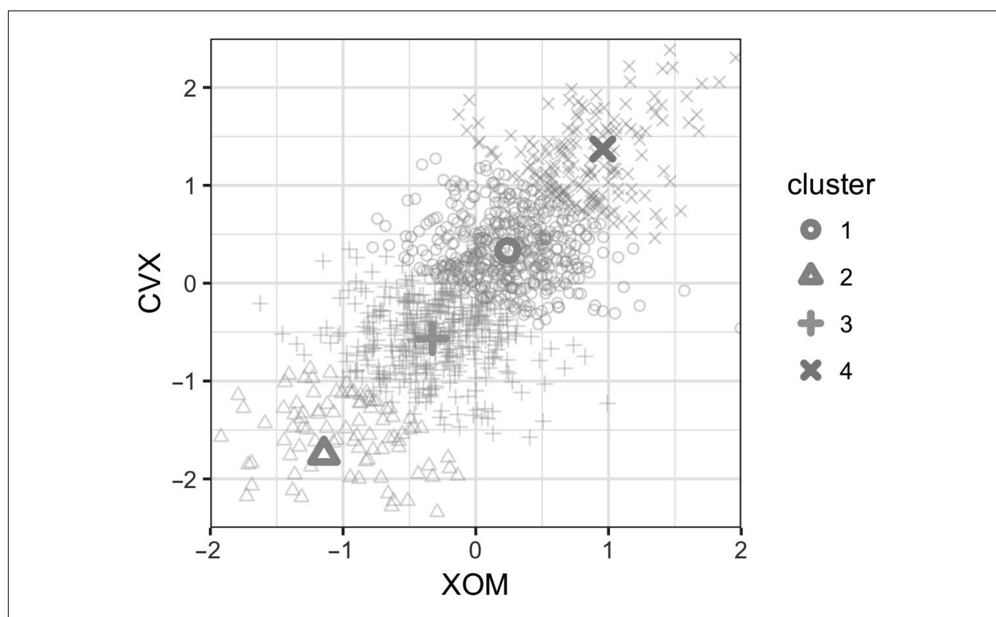


图 7-4：应用于 XOM 和 CVX 股价数据的 *K*-Means 聚类结果（稠密区域的两个类中心难以区分）

7.2.2　*K*-Means算法

K-Means 算法同样可以应用于具有 p 个变量 $(X_1, \cdots, X_p)$ 的数据集。要让 *K*-Means 给出精确解，计算难度很大，但启发式算法可以高效地计算出局部最优解。

在算法开始时，用户需要指定 K 值和一组初始的类均值，然后重复执行以下步骤。

(1) 根据距离的平方值，将每条记录分配给最近的类均值所在的类。
(2) 根据记录的分配情况，重新计算新的类均值。

一旦记录到类的分配情况不再改变，该算法就收敛。

在开始首次迭代前，需要指定一组初始的类均值。一般做法是将每个记录随机分配给 K 个类中的一个，然后计算类均值。

由于该算法并不保证能给出最优解，所以推荐做法是在初始化时使用不同的随机样本多次运行算法。当使用了多组迭代时，*K*-Means 的结果由类内平方和最低的一组迭代给出。

可以通过设置 R 函数 `kmeans` 的 `nstart` 参数，指定随机启动初始化的尝试次数。例如，下面的代码使用 10 个不同的初始类均值运行 *K*-Means，以找出 5 个类。

```
syms <- c( 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX', 'XOM', 'SLB', 'COP',
           'JPM', 'WFC', 'USB', 'AXP', 'WMT', 'TGT', 'HD', 'COST')
df <- sp500_px[row.names(sp500_px)>='2011-01-01', syms]
km <- kmeans(df, centers=5, nstart=10)
```

`kmeans` 函数会自动返回 10 个不同起始点中的最优解。我们可以通过设置参数 `iter.maxto`，指定每次随机启动所允许的最大迭代次数。

7.2.3　解释类

聚类分析的一个重要部分是解释类。在函数 `kmeans` 的输出中，最重要的两个是类规模和类均值。对于上节的例子，生成的类规模由下面的 R 命令给出。

```
km$size
[1] 186 106 285 288 266
```

类的规模相对平衡。不平衡的类可能是由过于离群的异常值所导致的，或是由于与其他数据迥异的一组记录所导致的，这两个问题都有可能导致需要进一步探索数据。

我们可以在 `ggplot` 中一并使用 `gather` 函数绘制类中心。

```
centers <- as.data.frame(t(centers))
names(centers) <- paste("Cluster", 1:5)
centers$Symbol <- row.names(centers)
centers <- gather(centers, "Cluster", "Mean", -Symbol)
centers$Color = centers$Mean > 0
ggplot(centers, aes(x=Symbol, y=Mean, fill=Color)) +
  geom_bar(stat='identity', position = "identity", width=.75) +
  facet_grid(Cluster ~ ., scales='free_y')
```

结果如图 7-5 所示，该图很好地揭示了各个类的本质。例如，类 1 和类 2 分别对应于股市下跌和上涨的交易日。类 3 和类 5 分别标识了消费类股票上涨的交易日和能源类股票下跌的交易日。类 4 表示了能源类股票上涨且消费类股票下跌的交易日。

图 7-5：各个类中的变量均值（即“类均值”）



聚类分析与主成分分析

对类均值绘图类似于查看主成分分析的载荷（参见 7.1.3 节）。二者的一个主要区别在于，类均值的符号是有意义的。主成分分析可以识别变异性的主要方向，而聚类分析可以发现彼此相邻的多组记录。

7.2.4 选择类的个数

要使用 *K*-Means 算法，必须指定类的个数 *K*。在一些情况下，类数是由应用决定的。例如，一家销售人员管理公司希望将客户群聚类为不同的“人物角色”，这样可以有针对性地引导电话销售。在这种情况下，管理上的考虑决定了所需的客户类数。如果设置两个类，可能无法反映出客户之间的差异；而如果设置 8 个类，可能会过多而难以管理。

如果没有实际的或管理上的考虑来决定类数，可以使用统计方法。并不存在一种可以给出“最佳”类数的标准方法。

一种常用的方法是“肘部法则”（elbow method），它能确定一组类何时解释了数据中的“大部分”方差。在这样一组类之上添加新的类，对于解释方差的贡献较小。“肘部”是指解释的累积方差在陡峭上升之后变为平整的转折点，方法也因此得名。

图 7-6 显示了类数从 2 增加到 15 时，针对默认数据解释的方差的累积百分比。那么该例的“肘部”在哪里？答案是没有明显的肘部，因为所解释的方差增量呈逐渐下降。如果数据中缺少良好定义的类，那么该问题是普遍存在的。这或许是肘部方法的一个缺点，尽管它的确能揭示数据的本质。

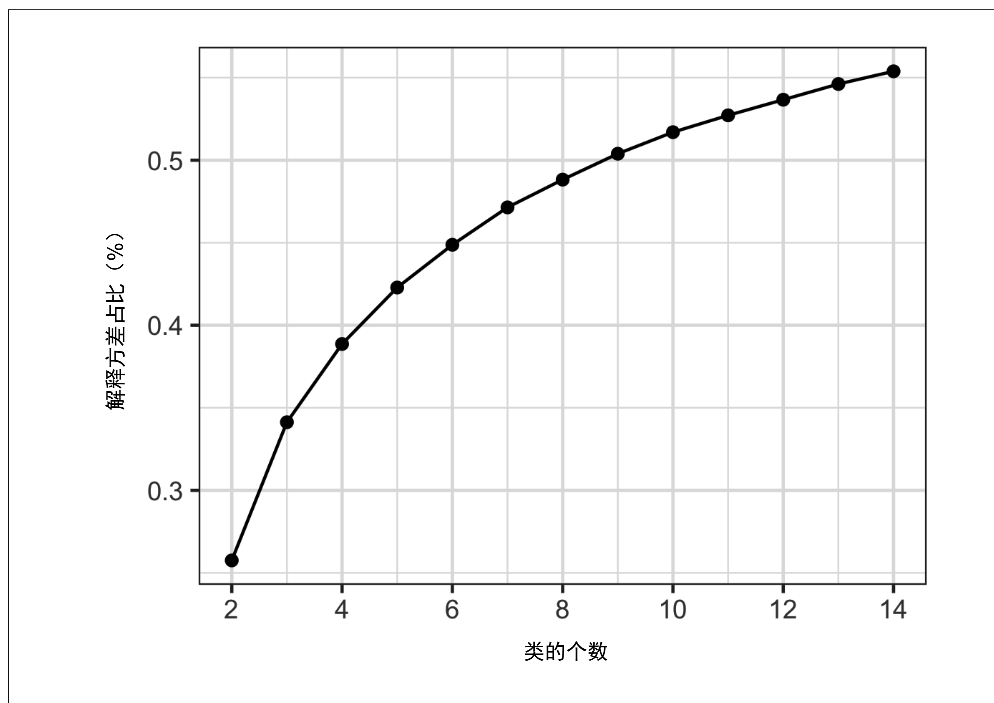


图 7-6：对股票数据应用肘部法则

R 语言的 `kmeans` 函数并没有提供单独的命令去应用肘部法则，但是肘部法则很容易应用于 `kmeans` 函数的输出，如下所示。

```
pct_var <- data.frame(pct_var = 0,
                      num_clusters=2:14)
totalss <- kmeans(df, centers=14, nstart=50, iter.max = 100)$totss
for(i in 2:14){
  pct_var[i-1, 'pct_var'] <- kmeans(df, centers=i, nstart=50, iter.max = 100)
    $betweenss/totalss
}
```

要评估应保留的类数，最重要的检验也许是：类是否可复用于新的数据？类是否可解释？类反映了数据的一般特性，还是仅仅反映了特定的实例？这在某种程度上可以使用交叉验证进行评估，参见 4.2.3 节。

一般来说，并不存在能可靠地指导所生成的类数的单一规则。



对于确定类数，存在多种更正式的方法，这些方法均基于统计学或信息理论。例如，罗伯特·蒂希雷尼、冈瑟·瓦尔特（Guenther Walther）和特雷弗·哈斯蒂基于统计理论提出了一种“间隙”（gap）统计量，用来识别肘部。但是对于大多数的应用而言，可能没有必要甚至不适合应用这些理论方法。

本节要点

- 所需的类数 *K* 由用户决定。
- *K*-Means 算法通过迭代地将记录分配给最近的类均值，直到类的分配情况不再发生改变，实现了类的生成。
- 通常，出于实际的考虑决定了 *K* 的选择。在统计学上不存在最优的类数。

7.3 层次聚类

除 *K*-Means 以外，**层次聚类**（hierarchical clustering）也是一种聚类方法，它可以生成非常不同的类。层次聚类比 *K*-Means 更灵活，并且更易于应用在非数值型变量上，对于发现离群的或异常的组和记录也更为敏感。层次聚类也适于做直观的图形展示，因而解释类也更为容易。

主要术语

树状图

一种可视化表示，显示了记录及其所属类的层次结构。

距离

测量两个**记录**之间的接近程度。

相异性

测量两个**类**之间的接近程度。

层次聚类的灵活性是有一定代价的，它不能很好地扩展到具有数百万条记录的大规模数据集上。即便是只有数万条记录的中等规模数据集，层次聚类可能也需要大量的计算资源。事实上，层次聚类的大部分应用都集中在一些规模相对较小的数据集上。

7.3.1　一个简单的例子

我们将层次聚类应用于一个具有 n 条记录和 p 个变量的数据集。其中，我们使用了两个基本度量。

- 距离度量 $d_{i,j}$ 测量两个记录 i 和 j 之间距离。
- 相异性度量 $D_{A,B}$，基于每个类内成员间的距离 $d_{i,j}$，测量两个类 A 和 B 间的差异。

对于使用数值型数据的应用，关键在于如何选择相异性度量。层次聚类首先使每个记录独自构成一个类，然后迭代地合并相异性最低的类。

在 R 语言中，可以使用 `hclust` 函数执行层次聚类。`hclust` 函数和 `kmeans` 函数的一大区别是，它并非运行在数据本身之上，而是运行于成对记录的距离 $d_{i,j}$ 之上。我们可以使用 `dist` 函数分别计算所有数据对间的距离。例如，下面的代码对一组公司的股票收益数据应用层次聚类。

```
syms1 <- c('GOOGL', 'AMZN', 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX',
           'XOM', 'SLB', 'COP', 'JPM', 'WFC', 'USB', 'AXP',
           'WMT', 'TGT', 'HD', 'COST')
# 下面执行转置操作。因为要按照公司聚类，所以需要股票数据按行排列
df <- t(sp500_px[row.names(sp500_px)>='2011-01-01', syms1])
d <- dist(df)
hcl <- hclust(d)
```

聚类算法将按记录（即 R 语言的 `dataframe` 对象的行）执行聚类。因为我们希望按公司进行聚类，所以要对 `dataframe` 对象做一次**转置**操作，使得股票数据按行排列，日期按列排列。

7.3.2　树状图

层次聚类可以天然地表示为树，我们称这样的绘图为**树状图**。树状图“dendrogram”一词源于希腊语“dendro”（树）和“gramma”（图）。在 R 语言中，使用 `plot` 命令可以很容易地生成一个树状图。

```
plot(hcl)
```

结果如图 7-7 所示。其中树的叶子对应于每条记录，分支的长度表示了相应类间的相异性程度。从图中可见，谷歌（GOOGL）和亚马逊（AMZN）股票的收益与其他股票的收益迥异。其他股票分成了自然组：能源类股票、金融类股票和消费类股票均划分成它们各自的子树。

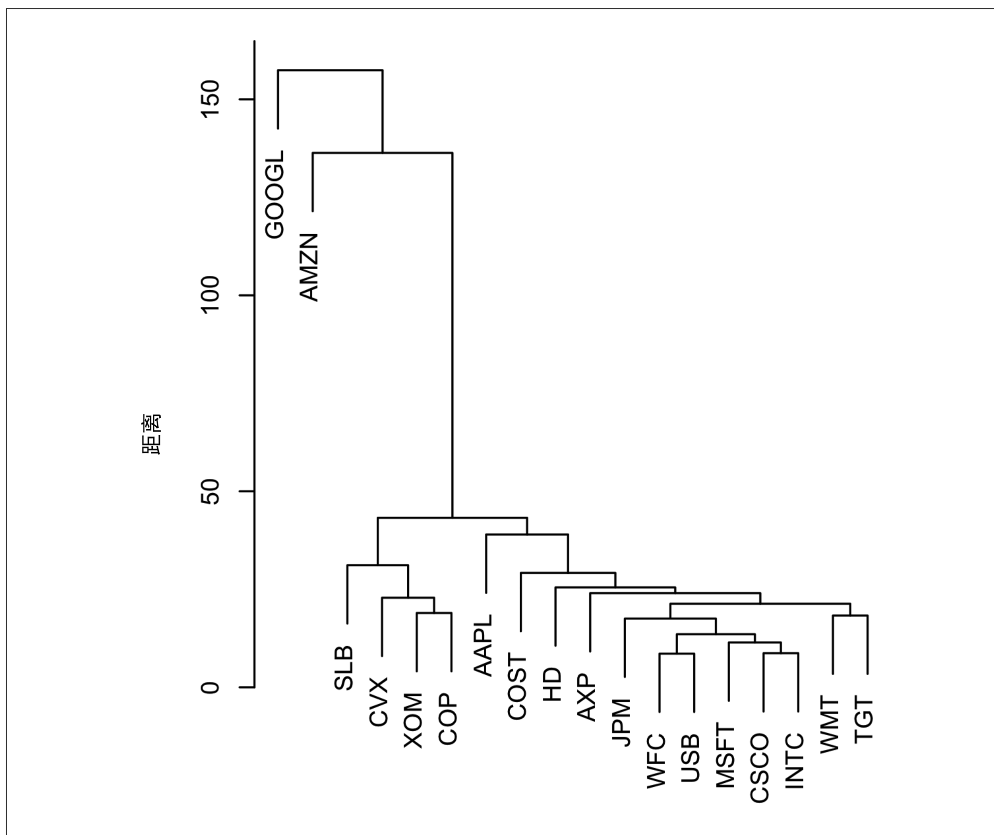


图 7-7：股票数据的树状图

不同于 *K*-Means，层次聚类不需要预先指定类数。如果要使用层级聚类划分指定数量的类，可以使用 `cutree` 函数。

```
cutree(hcl, k=4)
GOOGL  AMZN  AAPL  MSFT  CSCO  INTC   CVX   XOM   SLB   COP   JPM   WFC
    1     2     3     3     3     3     4     4     4     4     3     3
  USB   AXP   WMT   TGT    HD  COST
    3     3     3     3     3     3
```

在上面的命令中，我们设置类数为 4。从结果中可以看到，谷歌和亚马逊的股票分属于各自的类，石油类股票 XOM、CVS、SLB 和 COP 同属于另一个类，而其余的股票则被划分在第 4 个类中。

7.3.3　凝聚算法

层次聚类的主要算法是**凝聚**（agglomerative）算法。该算法迭代地合并相似的类。它从每条记录独自构成一个类开始，逐步地构建更大的类。该算法的第一步是计算所有记录对之间的距离。

对于每对记录 $(x_1, x_2,\cdots, x_p)$ 和 $(y_1, y_2,\cdots, y_p)$，凝聚算法使用距离度量（参见 6.1.2 节）测定两个记录间的距离 $d_{x,y}$。例如，我们可以使用欧氏距离：

$$d(x,y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_p-y_p)^2}$$

下面看一下如何计算类之间的距离。给定两个类 A 和 B，每个类中包含了一组不同的记录，$A = (a_1, a_2,\cdots, a_m)$ 和 $B = (b_1, b_2,\cdots, b_q)$。类之间的相异性 $D_{A,B}$ 可以通过类 A 和类 B 成员间的距离来测量。

一种相异性测量方法是使用**完全连接**（complete linkage）法。该方法使用了类 A 和类 B 之间所有记录对间的最大距离。

$$D(A,B)=\max d(a_i,b_j),\ a_i\in A,\ b_j\in B$$

完全连接法将相异性定义为所有记录对间的最大差异。

凝聚算法的主要步骤如下。

(1) 创建一组初始的类 C，数据中的每条记录各自构成一个类。
(2) 对于所有的 $C_K, C_L\in C$，计算成对类的相异性 $D(C_K, C_L)$。
(3) 合并由 $D(C_K, C_L)$ 测定的相异性最小的两个类 C_K 和 C_L。
(4) 如果剩余的类数不止一个，那么返回步骤 2。否则，终止算法。

7.3.4 测量相异性

测量相异性有四种常用的方法，即**完全连接**、**单一连接**、**平均连接**和**最小方差**。大部分层次聚类软件，包括前文介绍的 `hclust` 函数，都支持这四种测量方法（以及一些其他测量方法）。上一节已经介绍了完全连接方法，该方法趋向于生成具有类似成员的类。单一连接方法使用两个集群的记录对间距离的最小值。

$$D(A,B)=\min d(a_i,b_j)$$

其中 i 和 j 表示类内的记录对。单一连接法是一种“贪心”方法，它可以生成包含完全不同元素的类。平均连接法使用所有距离对的均值，它是单一连接和完全连接的一种折中。最小方差法（也被称为“Ward 方法”）类似于 K-Means 方法，因为它最小化类内平方和(参见 7.2 节)。

图 7-8 分别展示了采用四种方法做层次聚类的输出情况，使用的数据是 XOM 和 CVX 公司的股票收益数据。对于每种测量方法，我们生成四个类。

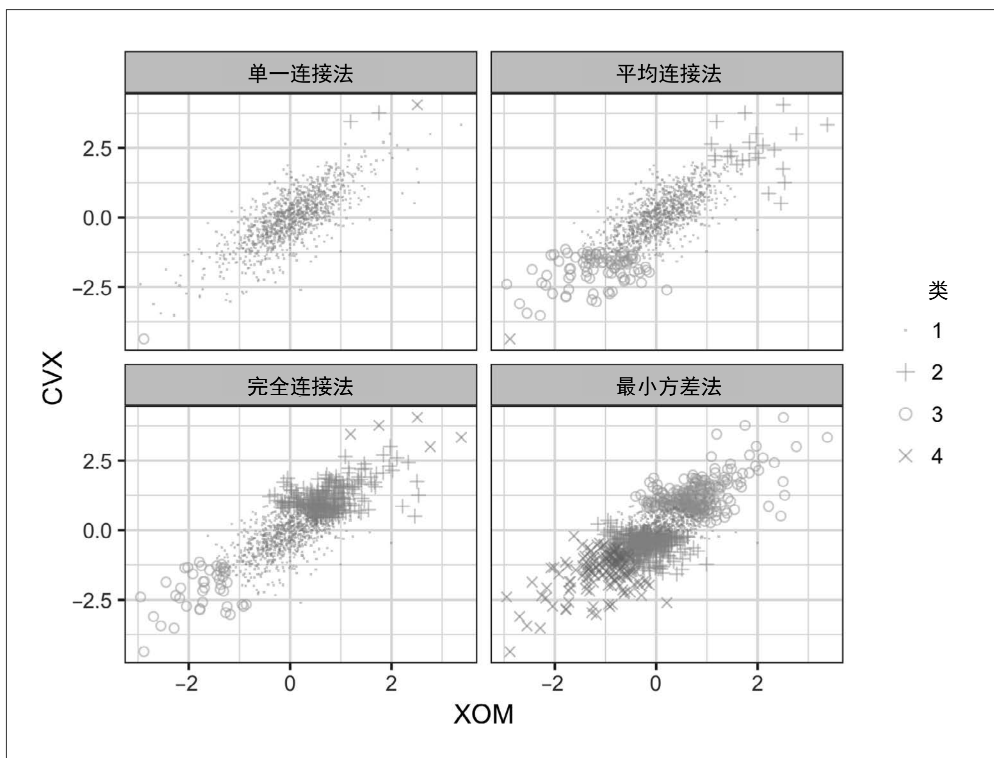


图 7-8：对于股票收益数据，比较不同的相异性测量

结果存在显著的差异。使用单一连接方法时，几乎所有的点都分配给同一个类。除了最小方差法之外，其他所有的测量方法都至少生成一个有少量离群值的类。如果与图 7-4 对比，我们可以看到，最小方差法与 *K*-Means 聚类最相似。

本节要点

- 层次聚类算法开始时，每条记录单独构成一个类。
- 在凝聚算法中，类逐步与相邻的类合并，直到所有记录属于单一类。
- 凝聚算法的类历史可以被保留并绘制出来。用户（无须预先指定类数）可以在算法执行的各个阶段，可视化地查看类数和类的结构。
- 有多种方法可以计算类之间的距离。这些方法都依赖于所有记录间距离。

7.4 基于模型的聚类

层次聚类、*K*-Means 等都是启发式聚类方法。这类方法主要依赖类内成员间距离的远近而发现类。它们直接测量数据，并不使用概率模型。在过去的 20 年间，研究人员为了开发出**基于模型的聚类**方法做出了大量努力。例如，华盛顿大学的 Adrian Raftery 等研究者对

基于模型的聚类方法做出了重大的理论上的和软件上的贡献。这种方法以统计理论为基础，为确定类的性质和数量提供了更严格的方法。这些方法可用于这种情况：一组记录彼此相似但并非相互接近（例如，收益差异很大的科技类股票），还有一组记录既是相似的也彼此接近（例如，收益差异较小的公用事业类股票）。

7.4.1 多元正态分布

最广为使用的基于模型的聚类方法依赖于**多元正态分布**。多元正态分布是对 p 个变量 X_1, X_2,…, X_p 正态分布的一种推广。该分布使用一组均值 $\mu = \mu_1, \mu_2, \cdots, \mu_p$ 和协方差矩阵 $\sum$ 定义。协方差矩阵是变量间相关性的度量（关于协方差的详细信息，参见 5.2.1 节）。协方差矩阵 $\sum$ 由 p 个方差 $\sigma_1^2, \sigma_2^2, \cdots, \sigma_p^2$ 以及所有变量对的协方差 $\sigma_{i,j}$（$i \neq j$）构成。矩阵的行和列均用变量表示，形式为：

$$\Sigma = \begin{vmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,p} \\ \sigma_{2,1} & \sigma_2^2 & \cdots & \sigma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p,1} & \sigma_{p,2}^2 & \cdots & \sigma_p^2 \end{vmatrix}$$

鉴于协方差矩阵是对称的，即 $\sigma_{i,j} = \sigma_{j,i}$，因此矩阵中只有 $p(p-1)-p$ 个协方差项，协方差矩阵共有 $p(p-1)$ 个参数。多元正态分布表示为：

$$(X_1, X_2, \cdots, X_p) \tilde{N}_p(\mu, \Sigma)$$

该符号化表示表明所有的变量均符合正态分布，整体分布使用变量均值的向量和协方差矩阵描述。

图 7-9 显示了具有两个变量 X 和 Y 的多元正态分布的概率轮廓线（例如，图中的 0.5 概率轮廓线包含了 50% 的分布）。

该分布的均值是 $\mu_x = 0.5$ 和 $\mu_y = -0.5$，协方差矩阵为：

$$\Sigma = \begin{vmatrix} 1 & 1 \\ 1 & 2 \end{vmatrix}$$

由于协方差 σ_{xy} 为正，所以 X 和 Y 是正相关的。

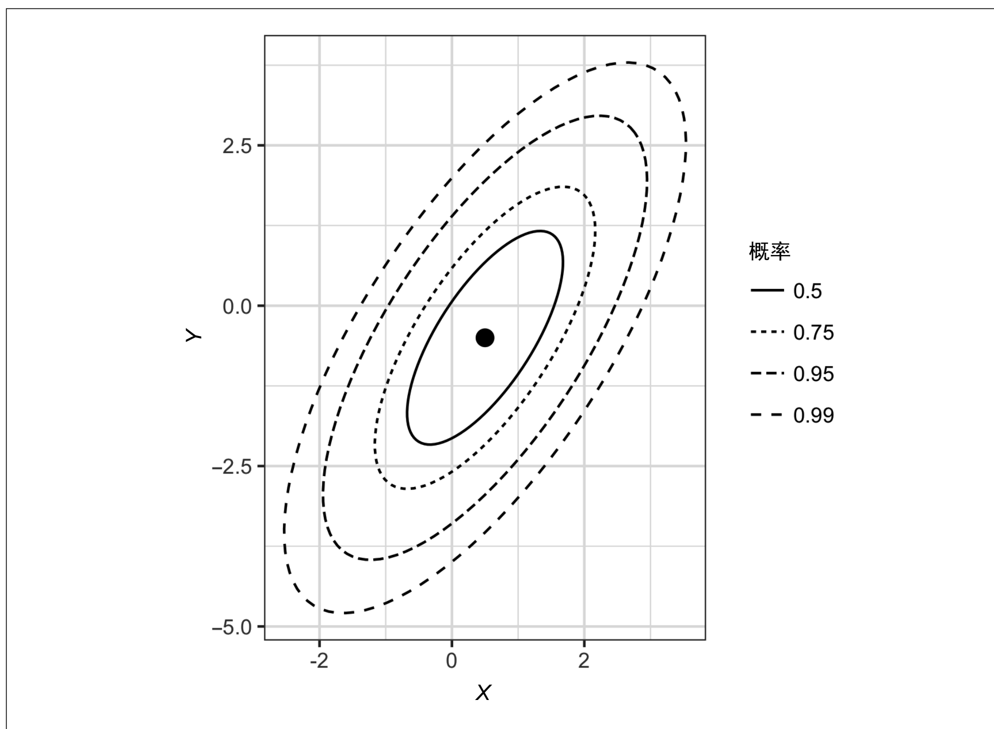


图 7-9：二维正态分布的概率轮廓线

7.4.2　混合正态分布

基于模型的聚类的关键思想是，假定每条记录的分布符合 K 个多元正态分布之一，其中 K 是类的个数。每个分布具有不同的均值 μ 和协方差矩阵 $\sum$。例如，对于两个变量 X 和 Y，每一行 (X_i,Y_i) 可建模为 K 个多元分布 $N_1(\mu_1, \sum_1), N_2(\mu_2, \sum_2), \cdots, N_K(\mu_K, \sum_K)$ 之一的一个抽样。

R 语言的 `mclust` 软件包提供了丰富的基于模型聚类的功能。该软件包最初是由 Chris Fraley 和 Adrian Raftery 开发的。我们可以对前面使用 K-Means 和层次聚类分析的股票收益数据，使用该软件包实现基于模型的聚类。

```
> library(mclust)
> df <- sp500_px[row.names(sp500_px)>='2011-01-01', c('XOM', 'CVX')]
> mcl <- Mclust(df)
> summary(mcl)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VEE (ellipsoidal, equal shape and orientation) model with 2 components:

 log.likelihood    n df       BIC       ICL
      -2255.134 1131  9 -4573.546 -5076.856
```

```
Clustering table:
  1   2 
963 168
```

在上面代码的执行过程中，我们可能会注意到，代码计算所用的时间明显长于其他过程。下面使用 predict 函数给出聚类情况，并绘图显示。

```
cluster <- factor(predict(mcl)$classification)
ggplot(data=df, aes(x=XOM, y=CVX, color=cluster, shape=cluster)) +
  geom_point(alpha=.8)
```

结果如图 7-10 所示。图中有两个类，一个类位于数据分布的中部，另一个类位于数据分布的外围。这完全不同于 *K*-Means（图 7-4）和层次聚类（图 7-8）的结果，这两种方法可以找到更紧凑的类。

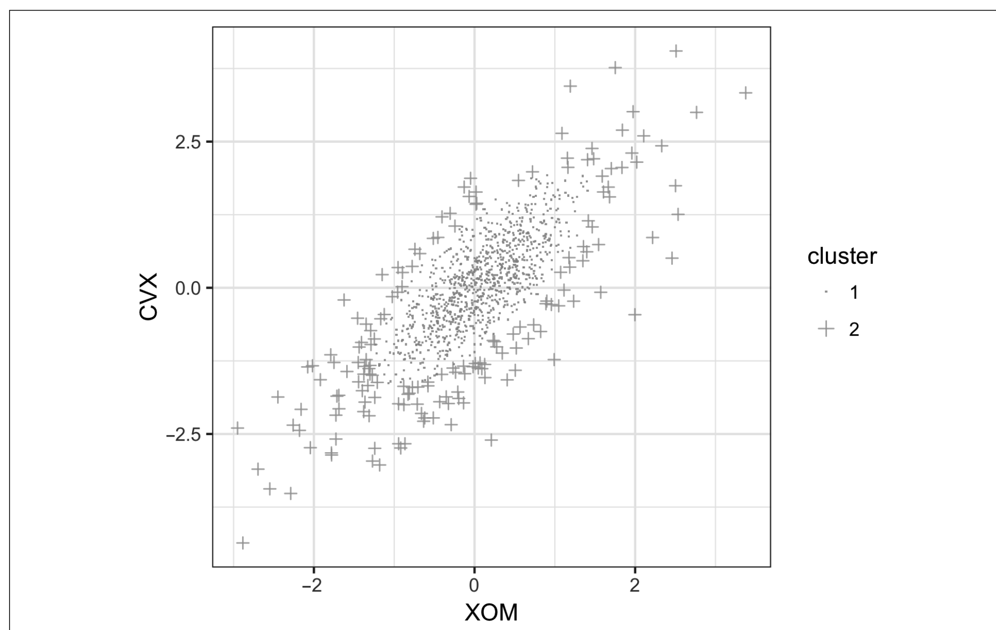


图 7-10：使用 mclust 得到股票收益数据的两个类

我们可以使用 summary 函数提取正态分布的参数。

```
> summary(mcl, parameters=TRUE)$mean
          [,1]        [,2]
XOM 0.05783847 -0.04374944
CVX 0.07363239 -0.21175715
> summary(mcl, parameters=TRUE)$variance
, , 1

          XOM       CVX
XOM 0.3002049 0.3060989
CVX 0.3060989 0.5496727
, , 2
```

```
          XOM      CVX
XOM 1.046318 1.066860
CVX 1.066860 1.915799
```

两个分布具有相似的均值和相关性，但第二个分布具有更大的方差和协方差。

`mclust` 给出的聚类可能令人惊奇，但实际上它们展示了这种方法的统计性质。基于模型聚类的目标是找到一组最佳拟合的多元正态分布。股票数据看上去具有正态分布的形状（参见图 7-9 中的轮廓线）。实际上，相比于正态分布，股票收益的分布具有更长的尾部。为了解决这个问题，`mclust` 对大量数据拟合了一个分布，随后又拟合了具有更大方差的第二个分布。

7.4.3 类数的选取

不同于 *K*-Means 和层次聚类，`mclust` 会自动选取类数（在本例中，是两个类）。这是通过选取使**贝叶斯信息准则**（BIC）值最大的类数实现的。BIC（类似于 AIC）是在一组候选模型中找到最佳模型的通用工具。例如，AIC（或 BIC）常用于在逐步回归（参见 4.2.4 节）中选择模型。BIC 通过对模型中的参数数量添加一个惩罚项，选择最优拟合模型。在基于模型的聚类中，增加类数总是会提高模型的拟合度，但代价是在模型中额外地引入了一些参数。

我们可以使用 `hclust` 软件包中的函数，绘制不同类数的 BIC 值。

```
plot(mcl, what='BIC', ask=FALSE)
```

结果如图 7-11 所示。图中，x 轴显示了类数，即不同的多元正态模型（成分）的数量。

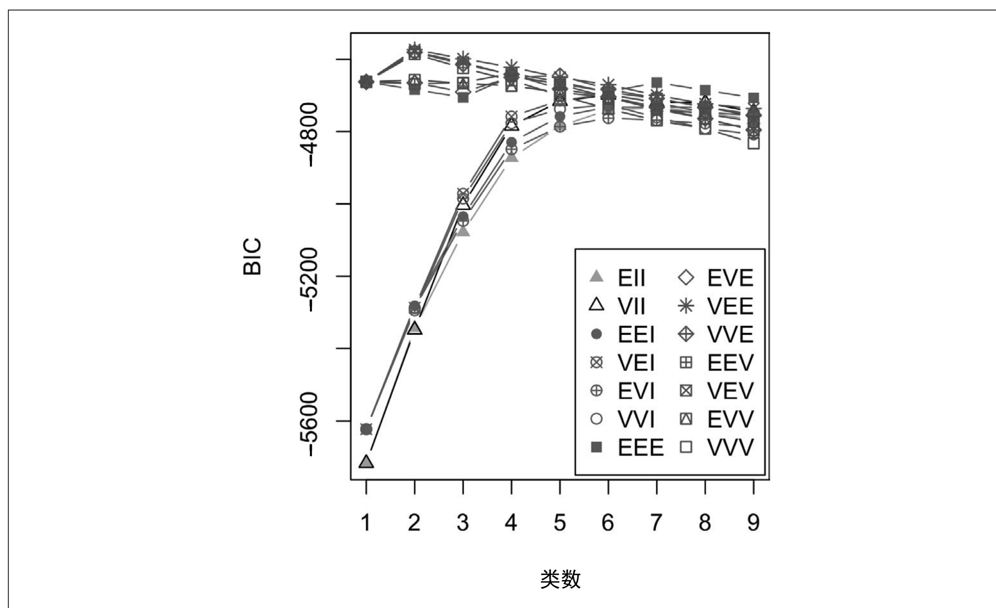


图 7-11：在选取不同类（成分）数的情况下，股票收益数据的 BIC 值

该绘图类似于确定 *K*-Means 的类数时所使用的肘部图（参见图 7-6），只是绘制的值是 BIC，而不是方差被解释的百分比。其中一个很大的差异在于，`mclust` 显示的并非一行，而是同时显示了 14 个不同的行！这是因为 `mclust` 实际上为每种聚类规模拟合了 14 个不同的模型，并最终选择一个最优拟合模型。

为什么 `mclust` 需要拟合这么多的模型，才能确定最优的多元正态分布？这是因为在拟合模型时，有多种方法可以参数化协方差矩阵 Σ。在大多数情况下，我们并不需要操心模型的细节，可以简单地使用 `mclust` 所选择的模型。在本例中，根据 BIC，三个不同的模型（称为 VEE、VEV 和 VVE）使用两个成分给出了最优拟合。



基于模型的聚类是一个正在快速发展的研究领域，本书只介绍了该领域中的一小部分内容。事实上，仅是 `mclust` 的帮助文件就长达 154 页。对于数据科学家碰到的大多数问题来说，无须耗费过多的精力去查看基于模型的聚类的细节。

基于模型的聚类技术确实存在一些限制。此类方法需要根据数据情况对模型做出一个潜在的假设，并且聚类的结果非常依赖于该假设。此类方法的计算资源需求甚至高于层次聚类，因此很难扩展到大数据上。此外，该算法比其他方法更复杂，更难以使用。

本节要点

- 基于模型的聚类方法假设类是由不同数据生成过程所生成的，各个数据生成过程具有不同的概率分布。
- 基于模型的聚类方法拟合了不同的模型,假设有不同数量的分布（通常是正态分布）。
- 基于模型的聚类方法无须使用过多的参数（即过拟合），就能选出一个能很好地拟合数据的模型（以及类数）。

7.4.4 拓展阅读

基于模型的聚类的更多信息，参见 `mclust` 的文档。

7.5 变量的缩放和分类变量

使用无监督学习技术时，一般需要适度地缩放数据。这不同于许多回归和分类技术（除了 *K* 最近邻算法外，参见 6.1 节），其中变量的缩放并不重要。

主要术语

缩放

缩小或放大数据的方法，常用于将多个变量缩放到同一尺度上。

归一化

一种通过减去均值并除以标准偏差进行缩放的方法。

同义词：标准化

高氏距离（Gower's distance）

一种应用于数值数据和类别数据相混合的缩放算法。它可以将所有变量缩放到 [0, 1] 范围内。

以个人贷款数据为例，数据中的变量在单位和数量上存在很大的差别。其中，一些变量的值相对较小，例如工作年限；而另一些变量的值则非常大，例如以美元为单位的贷款数额。如果不缩放数据，那么主成分分析、*K*-means 等聚类方法将会被数值较大的变量所主导，进而忽略那些数值较小的变量。

在一些聚类过程中，分类数据可能会引发一些特殊的问题。就 *K* 最近邻而言，无序因子变量通常需要使用独热编码（参见 6.1.3 节）转换为一组值为 0 或 1 的二元变量。而二元变量不仅会与其他数据具有不同的规模，而且二元变量只有两个值这一事实就可以证明主成分分析、*K*-means 等方法会存在问题。

7.5.1 变量的缩放

在应用聚类之前，需要对具有完全不同尺度和单位的变量做适当的归一化处理。例如，下面代码不对贷款拖欠数据做归一化处理，就应用了 kmeans 函数。

```
df <- defaults[, c('loan_amnt', 'annual_inc', 'revol_bal', 'open_acc',
                   'dti', 'revol_util')]
km <- kmeans(df, centers=4, nstart=10)
centers <- data.frame(size=km$size, km$centers)
round(centers, digits=2)
  size loan_amnt annual_inc revol_bal open_acc   dti revol_util
1    55  23157.27  491522.49  83471.07    13.35  6.89      58.74
2  1218  21900.96  165748.53  38299.44    12.58 13.43      63.58
3  7686  18311.55   83504.68  19685.28    11.68 16.80      62.18
4 14177  10610.43   42539.36  10277.97     9.60 17.73      58.05
```

从结果中看到，在类中占主导地位的是变量 annual_inc 和 revol_bal，而且类规模的差异也很大。类 1 只有 55 个成员，即那些具有较高收入和循环信贷的账户。

一种常用的缩放变量方法是归一化或标准化（参见 6.1.4 节）。归一化将数据减去平均值并除以标准偏差，转换为 z 分数。有关使用 z 分数的更多介绍，参见 6.1.4 节。

$$z = \frac{x - \overline{x}}{s}$$

下面，我们将 kmeans 应用于归一化的数据，再次查看类所发生的变化。

```
df0 <- scale(df)
km0 <- kmeans(df0, centers=4, nstart=10)
centers0 <-scale(km0$centers, center=FALSE,
                 scale=1/attr(df0, 'scaled:scale'))
centers0 <- scale(centers0, center=-attr(df0, 'scaled:center'), scale=F)
data.frame(size=km0$size, centers0)
```

```
  size loan_amnt annual_inc revol_bal open_acc   dti revol_util
1 5429  10393.60   53689.54   6077.77     8.69 11.35      30.69
2 6396  13310.43   55522.76  16310.95    14.25 24.27      59.57
3 7493  10482.19   51216.95  11530.17     7.48 15.79      77.68
4 3818  25933.01  116144.63  32617.81    12.44 16.25      66.01
```

从这次的结果中可以看到，类的规模更加平衡。类不再仅被 annual_inc 和 revol_bal 所主导，而是更多地揭示了数据中有意义的结构信息。注意，类的中心要重新缩放回前面代码中使用的原始单位。如果不做缩放，那么结果值会以 *z* 分数给出，这会降低结果的可解释性。



缩放对于主成分分析同样十分重要。在计算主成分时，使用 *z* 分数等同于使用相关矩阵（参见 1.7 节），而不是等同于使用协方差矩阵。计算主成分分析的软件中通常会提供设置使用相关矩阵的选项，例如在 R 语言中，可以指定 princomp 函数的参数 cor。

7.5.2　控制变量

即便在测量中所有变量使用了同一尺度，也准确地反映了相对重要性（例如股票价格的变动），有时还是需要重新缩放变量。

下面，我们将 Alphabet（GOOGL）和亚马逊（AMZN）的股票添加到 7.1.3 节所做的分析中。

```
syms <- c('AMZN', 'GOOGL' 'AAPL', 'MSFT', 'CSCO', 'INTC', 'CVX', 'XOM',
   'SLB', 'COP', 'JPM', 'WFC', 'USB', 'AXP', 'WMT', 'TGT', 'HD', 'COST')
top_sp1 <- sp500_px[row.names(sp500_px)>='2005-01-01', syms]
sp_pca1 <- princomp(top_sp1)
screeplot(sp_pca1)
```

生成的陡坡图如图 7-12 所示。我们知道，陡坡图可以显示第一主成分的方差。从图 7-12 中可以看到，第一主成分和第二主成分间的差异远大于其他的主成分。这通常表明载荷是由一两个变量支配的。在下面的例子中我们可以看出情况的确如此。

```
round(sp_pca1$loadings[,1:2], 3)
      Comp.1 Comp.2
GOOGL  0.781  0.609
AMZN   0.593 -0.792
AAPL   0.078  0.004
MSFT   0.029  0.002
CSCO   0.017 -0.001
INTC   0.020 -0.001
CVX    0.068 -0.021
XOM    0.053 -0.005
SLB    0.079 -0.013
…
```

第一主成分和第二主成分几乎完全由 GOOGL 和 AMZN 所支配。这是因为 GOOGL 和 AMZN 股票的价格走势主导着变异性。

要处理这种情况，我们可以原封不动地添加数据，并重新缩放变量（参见 7.5.1 节），也可以在分析中剔除控制变量，并对这些控制变量做单独处理。并不存在一个所谓“正确”的

方法，具体使用的处理方法取决于应用。

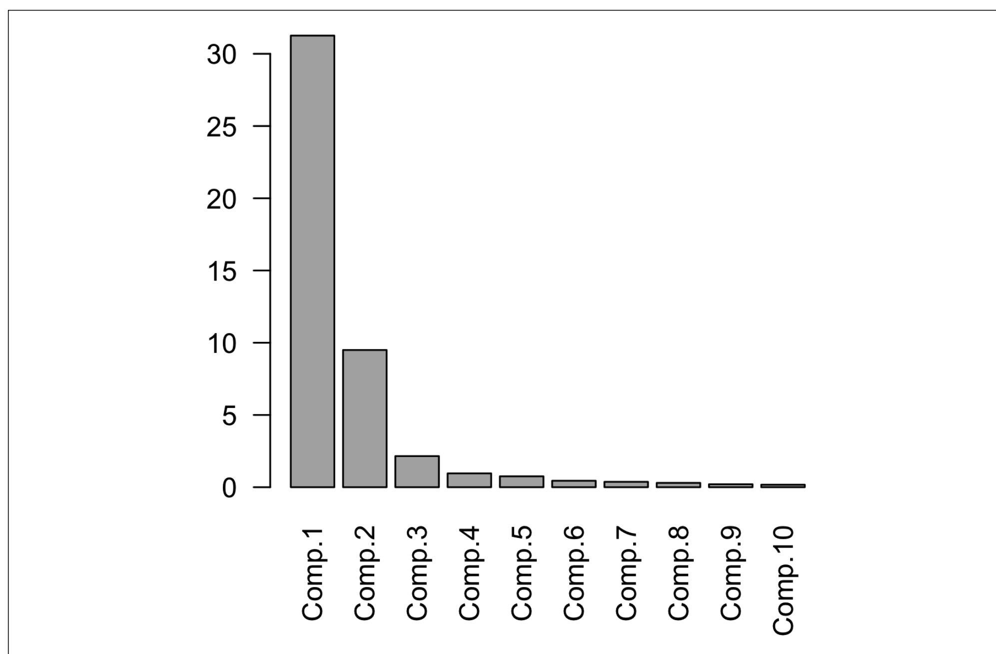


图 7-12：添加 GOOGL 和 AMZN 股票后，标准普尔 500 指数中排名靠前的几家公司股票的主成分分析的陡坡图

7.5.3 分类数据和高氏距离

分类数据必须要转换为数值型数据。转换可以通过排序（对于有序因子）实现，也可以通过编码为一组二元变量（虚拟变量）实现。如果数据中有连续变量和二元变量，那么一般需要缩放变量，使各个变量间具有相似的范围（参见 7.5.1 节）。一种常用方法是使用**高氏距离**。

高氏距离的基本思想是，根据数据类型，对每个变量应用不同的距离度量。

- 对于数值型变量和有序因子，高氏距离计算为两条记录间差异的绝对值（即**曼哈顿距离**）。
- 对于分类变量，如果两个记录属于不同的类，那么距离为 1；如果它们属于同一个分类，那么距离为 0。

高氏距离的计算步骤如下。

(1) 对每条记录，计算所有变量对 i 和 j 间的距离 $d_{i,j}$。
(2) 将每个距离 $d_{i,j}$ 缩放到区间 [0, 1] 中。
(3) 使用简单均值或加权均值，将所有变量对间的缩放距离相加，创建一个距离矩阵。

为了更好地解释高氏距离，我们以贷款数据为例，从数据中取几行。

```
> x = defaults[1:5, c('dti', 'payment_inc_ratio', 'home', 'purpose')]
> x
# A tibble: 5 × 4
    dti payment_inc_ratio   home            purpose
  <dbl>             <dbl> <fctr>             <fctr>
1  1.00           2.39320   RENT                car
2  5.55           4.57170    OWN     small_business
3 18.08           9.71600   RENT              other
4 10.08          12.21520   RENT debt_consolidation
5  7.06           3.90888   RENT              other
```

然后使用 cluster 软件包中的 daisy 函数计算高氏距离。

```
> library(cluster)
> daisy(x, metric='gower')
Dissimilarities :
          1         2         3         4
2 0.6220479
3 0.6863877 0.8143398
4 0.6329040 0.7608561 0.4307083
5 0.3772789 0.5389727 0.3091088 0.5056250
```

从结果中可以看到，所有的距离值都介于 0 和 1 之间。距离最大的记录对是记录 2 和记录 3。这两条记录中的数值型变量 home 或 purpose 的值不同，分类变量 dti（债务与收入比）和 payment_inc_ratio 的层级也完全不同。记录 3 和记录 5 之间的距离最小，因为它们的 home 或 purpose 变量具有相同的值。

下面对 daisy 函数的输出调用 hclust，对所生成的距离矩阵应用层次聚类（参见 7.3 节）。

```
df <- defaults[sample(nrow(defaults), 250),
               c('dti', 'payment_inc_ratio', 'home', 'purpose')]
d = daisy(df, metric='gower')
hcl <- hclust(d)
dnd <- as.dendrogram(hcl)
plot(dnd, leaflab='none', ylab='distance')
```

结果生成如图 7-13 所示的树状图。从图中可以看到，单个记录在 x 轴上是不可区分的，但是可以使用下面的代码检查其中某个子树中的记录。本例中指定了左边的子树，使用的“截止值”为 0.5。

```
> df[labels(dnd_cut$lower[[1]]),]
# A tibble: 9 × 4
    dti payment_inc_ratio   home purpose
  <dbl>             <dbl> <fctr>  <fctr>
1 24.57           0.83550   RENT   other
2 34.95           5.02763   RENT   other
3  1.51           2.97784   RENT   other
4  8.73          14.42070   RENT   other
5 12.05           9.96750   RENT   other
6 10.15          11.43180   RENT   other
7 19.61          14.04420   RENT   other
8 20.92           6.90123   RENT   other
9 22.49           9.36000   RENT   other
```

该子树完全由贷款用途为“其他”的借款者组成。虽然无法真正地实现所有子树的严格分

离，但这已经说明了分类变量趋向于聚集在类中。

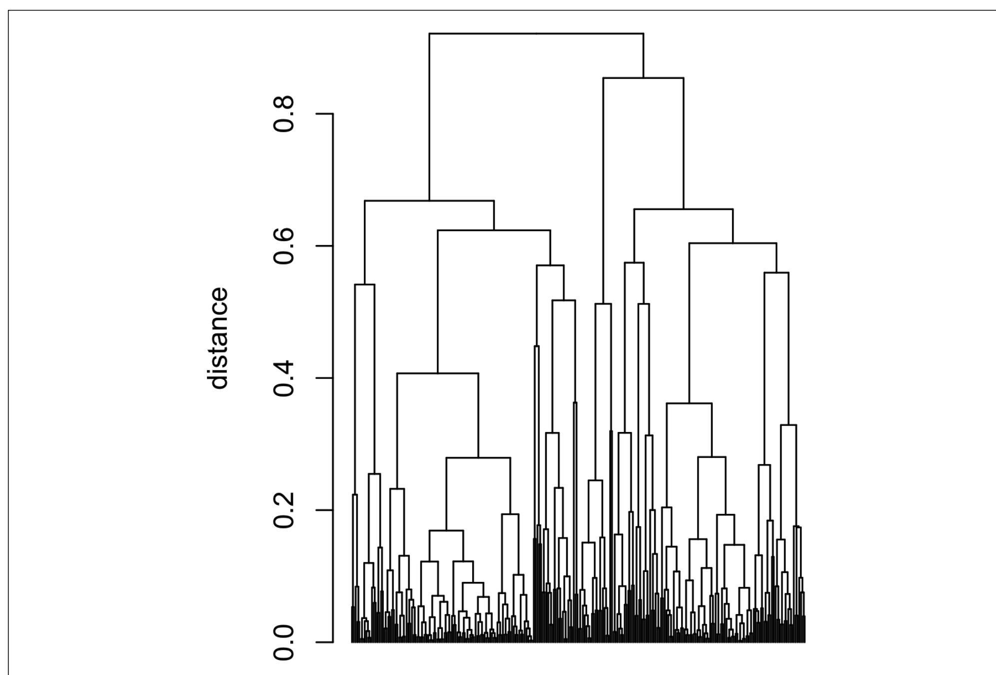


图 7-13：对具有多种类型变量的贷款拖欠数据样本应用 hclust 所生成的树状图

7.5.4 混合数据的聚类问题

K-Means 和主成分分析最适合用于连续变量。对于较小的数据集，使用基于高氏距离的层次聚类更好。从原理上看，*K*-Means 完全适用于二元数据和分类数据。我们通常会使用"独热编码"（参见 6.1.3 节），将分类数据转换为数值型数据。然而在实践中，很难对二元数据应用 *K*-Means 和主成分分析。

如果使用标准的 *z* 分数，那么二元变量将会主导聚类的定义。这是因为 0/1 变量只有两个值，而 *K*-Means 是通过将所有的 0 或 1 记录指定给聚类，获得较小的类内平方和。例如，下面我们将 kmeans 函数应用于具有因子变量 home 和 pub_rec_zero 的贷款拖欠数据。

```
df <- model.matrix(~ -1 + dti + payment_inc_ratio + home + pub_rec_zero,
                   data = defaults)
df0 <- scale(df)
km0 <- kmeans(df0, centers=4, nstart=10)
centers0 <-scale(km0$centers, center=FALSE,
                 scale=1/attr(df0, 'scaled:scale'))
round(scale(centers0, center=-attr(df0, 'scaled:center'), scale=False), 2)
    dti payment_inc_ratio homeMORTGAGE homeOWN homeRENT pub_rec_zero
1 17.02              9.10         0.00       0     1.00         1.00
2 17.47              8.43         1.00       0     0.00         1.00
3 17.23              9.28         0.00       1     0.00         0.92
4 16.50              8.09         0.52       0     0.48         0.00
```

从结果中可以看出，因子变量的不同层级基本上由前 4 个类所代理。为了避免出现这样的问题，我们可以对二元变量做缩放，使其具有比其他变量更小的方差。对于非常大的数据集，我们可以将聚类应用于具有不同分类值的各个数据子集。例如，我们可以将聚类分别应用于有房贷的借款者、完全拥有房屋的借款者或租赁房屋的借款者。

本节要点

- 以不同尺度测量的变量，需要转换到相似的尺度上。这样，变量对算法的影响不会主要由变量的尺度决定。
- 归一化（标准化）是一种常用的缩放方法——减去均值再除以标准偏差计算。
- 另一种缩放方法是高氏距离，它将所有的变量缩放到 [0, 1] 范围内。高氏距离通常用于含有数值型数据和分类数据的混合数据。

7.6 小结

对于数值型数据的降维，主要使用的工具是主成分分析和 *K*-Means 聚类。使用这两种方法时，需要适当地缩放数据，以确保数据的归约有意义。

如果对聚类良好分离的高度结构化数据做聚类，所有方法可能会给出相似的结果。不同的方法各有优点。*K*-Means 可以扩展到规模非常大的数据，结果也易于理解。层次聚类可应用于含有数值型数据和分类数据的混合数据，并能给出直观的显示（即树状图）。基于模型的聚类方法不同于上面两种启发式方法，它是根据统计学理论提出的，因此更为严格。但是对于规模非常大的数据集，聚类主要使用 *K*-Means 方法。

对于有噪声的数据，例如贷款数据和股票数据（数据科学家所面对的大部分数据集都是有噪声的），选择更为重要。*K*-Means、层次聚类，尤其是基于模型的聚类，都会给出完全不同的解决方案。那么数据科学家应该如何选择呢？不幸的是，没有一种简单的经验法则可提供指导。最终选用哪种方法，取决于数据的规模和应用的目标。

作者简介

彼得·布鲁斯（Peter Bruce）是 Statistics.com 统计学教育学院的创立者，并推动其发展壮大。该学院目前提供约 100 门统计学课程，其中近三分之一的课程是专门面向数据科学家的。在聘请顶尖作者担任讲师，以及制定学院营销策略，以对接专业数据科学家的过程中，Peter 不仅广泛地了解了目标市场情况，而且也丰富了自己的统计专业知识。

安德鲁·布鲁斯（Andrew Bruce）在学术界、政府和企业中具有 30 多年的统计学和数据科学经验。他获得了华盛顿大学的统计学博士学位，并在学术期刊上发表了大量的论文。他曾针对从成熟的金融公司到互联网创业公司等多个行业面对的多种问题，提出了基于统计学的解决方案。他对数据科学实践也有着深刻的见解。

封面说明

本书封面上的动物是一种粗腿厚纹蟹（学名：Pachygrapsus crassipes），也称为条纹岸蟹，分布于北美洲、中美洲、韩国和日本的太平洋沿岸海滩。这种甲壳类动物生活在岩石、潮汐池和裂缝中，一生中有大约一半的时间在陆地上，但会定期返回到水中，以保持鳃的湿润。

条纹岸蟹的命名源于其褐黑色甲壳上的绿色条纹。它们的蟹爪是红色的，腿是紫色的，上面也有条纹或斑驳图案。它们一般可以长到 3 至 5 厘米，雌性略小。它们的眼睛呈秆状，可以灵活旋转，这使它们在行走时具有完整的视野。

螃蟹是一种杂食动物，主要摄食藻类，但也食用软体动物、蠕虫、真菌、死亡动物和其他甲壳动物，具体取决于可获取的食物。它们在发育成熟的过程中要多次换壳，通过吸入水分来扩张并打开旧壳。一旦完成换壳，它们要经历数小时的艰难时光方可自由行动。这时它们必须隐藏起来，直到新壳完全硬化。

O’Reilly 图书封面上的许多动物正濒临灭绝。这些动物对于我们这个世界非常重要。要详细了解如何帮助它们，请访问 animals.oreilly.com。

封面图片来自 *Pictorial Museum of Animated Nature*。



微信连接



回复“数据科学”查看相关书单



微博连接

关注@图灵教育 每日分享IT好书



QQ连接

图灵读者官方群I：218139230
图灵读者官方群II：164939616

图灵社区
iTuring.cn

在线出版，电子书，《码农》杂志，图灵访谈

O'REILLY®

面向数据科学家的实用统计学

统计学方法是数据科学的重要组成部分，但极少有数据科学家接受过正规的统计学教育或培训，而关于统计学基础的课程和教材也很少从数据科学的角度进行讲解。本书专门从数据科学的角度阐释重要且实用的统计学概念，重点介绍如何将各种统计学方法应用于数据科学。

- 为什么探索性数据分析是数据科学关键的第一步
- 随机抽样如何降低偏差、生成高质量数据集
- 实验设计原则如何针对问题生成确定性答案
- 如何使用回归方法估计结果并检测异常
- 用于预测记录所属类别的主要分类方法
- 从数据中“学习”的统计机器学习方法
- 从未标记数据中提取有意义信息的无监督学习方法

“本书并非又一本统计学教程，也不是机器学习手册。它运用清晰的解释和丰富的示例，将实用的统计学术语与当下数据挖掘的行话和实践联系起来。对数据科学入门者和经验丰富的数据科学从业者来说，这都是一本非常出色的参考书。”

——**Galit Shmueli**
知名数据挖掘学者

彼得·布鲁斯（Peter Bruce），知名统计学家，Statistics.com统计学教育学院的创立者兼院长，重采样统计软件的开发者。曾在美国马里兰大学和各种短训班教授重采样统计课程。

安德鲁·布鲁斯（Andrew Bruce），华盛顿大学统计学博士，拥有30多年的统计学和数据科学经验，在多家知名学术期刊上发表过多篇论文。

封面设计：Karen Montgomery 张健

图灵社区：iTuring.cn
热线：(010)51095186转600



分类建议 计算机／统计学

人民邮电出版社网址：www.ptpress.com.cn

O'Reilly Media, Inc.授权人民邮电出版社出版

此简体中文版仅限于中国大陆（不包含中国香港、澳门特别行政区和中国台湾地区）销售发行
This Authorized Edition for sale only in the territory of People's Republic of China
(excluding Hong Kong, Macao and Taiwan)



ISBN 978-7-115-49366-8
定价：89.00元

看完了

如果您对本书内容有疑问，可发邮件至 contact@turingbook.com，会有编辑或作译者协助答疑。也可访问图灵社区，参与本书讨论。

如果是有关电子书的建议或问题，请联系专用客服邮箱：ebook@turingbook.com。

在这可以找到我们：

微博 @图灵教育：好书、活动每日播报
微博 @图灵社区：电子书和好文章的消息
微博 @图灵新知：图灵教育的科普小组
微信 图灵访谈：ituring_interview，讲述码农精彩人生
微信 图灵教育：turingbooks